STOCHASTIC RECONSTRUCTION OF SNOW MICROSTRUCTURE FROM X-RAY MICROTOMOGRAPHY IMAGES

By

Hongyan Yuan

RECOMMENDED:

Advisory Committee Chair

Chair, Department of Mechanical Engineering

APPROVED:

Dean, College of Engineering and Mines

Dean of the Graduate School

Aug 15, 2007

Date

STOCHASTIC RECONSTRUCTION OF SNOW MICROSTRUCTURE FROM X-RAY MICROTOMOGRAPHY IMAGES

A

THESIS

Presented to the Faculty

of the University of Alaska Fairbanks

in Partial Fulfillment of the Requirements

for the Degree of

MASTER OF SCIENCE

By

Hongyan Yuan, B.Eng.

Fairbanks, Alaska

August 2007

ABSTRACT

The three-dimensional (3D) high-resolution digitized snow microstructure (pixel size 6 micron) was obtained by X-ray microtomography. The experimental result was verified by measuring the density of the snow sample. Statistical characteristics (porosity, local porosity, two-point correlation function) were extracted from cross-sectional images. The one-level-cut Gaussian random field model was used to stochastically reconstruct snow microstructure from X-ray microtomography images. Efficient computer programs were developed in MATLAB for the whole stochastic reconstruction procedure, including the numerical inversion of the correlation function and the generation of 3D large-scale Gaussian random fields by 3D inverse fast Fourier transform. The quality of the reconstruction was assessed by comparing the two-point correlation function and cross-sectional images.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

LIST OF APPENDICES

NOMENCLATURE

p_{expt}	Measured porosity
$p_{\text{expt}}^{(2)}(\mathbf{r})$	Measured two-point correlation function
$p_{\text{expt}}^{(2)}(r_i)$	Measured isotropic two-point correlation function
$M(\mathbf{r}, H)$	Cubic measurement cell
$\psi(\mathbf{r}, H)$	Local porosity
$y(\mathbf{r})$	Gaussian random field
$g(r)$	Correlation function
$\phi(\mathbf{r})$	Phase function
α	One-level-cut parameter
$p^{(n)}$	n-point correlation function
p	Porosity
$p^{(2)}$	Two-point correlation function
$\rho(k)$	Power spectral density function

ACKNOWLEDGEMENTS

Most of all, I wish to express my deepest gratitude to Dr. Jonah Lee, whom I was lucky to have as an advisor. This thesis would have not been possible without his support. I would also like to thank Dr. Cheng-fu Chen and Dr. Jing Zhang for being my committee members. Additionally, I would like to thank Eric Johansen for making snow sample holders for the X-ray microtomography experiment.

This study was financially supported by the Alaska Cold Region Automotive Research Group at the University of Alaska Fairbanks. I highly appreciate their financial support.

I also thank ARSC for using their resources for our computations. I thank the consultants of ARSC for their help.

I thank all my friends for their help and affection. Finally, I would like to express my gratitude to all those who helped me complete my thesis.

Chapter 1 INTRODUCTION

Snow mechanics is the theoretical and applied science of the mechanical behavior of snow; it is that branch of mechanics concerned with the response of snow to the force fields of its environment [Shapiro, Johnson et al., 1997]. The problems associated with snow mechanics in geophysics and engineering, such as avalanche mechanics [Lehning, Bartelt et al., 2002] and tire-snow interaction [Shoop, 2001; Lee and Liu, 2005; Zhang, Lee et al., 2005; Klein-Paste, Sinha et al., 2007] and other applications have attracted considerable attention and interest among engineers and researchers However, it has proved difficult to solve these problems due to the strong dependence of snow macroscopic properties on the bonding, microstructure, and the geometric characteristics of the grains [Shapiro, Johnson et al., 1997; Johnson and Hopkin, 2005]. A crucial problem for snow is to obtain a quantitative geometric three-dimensional (3D) description of its microstructure.

There are several ways to obtain snow microstructure experimentally. The thin sectioning technique was used to obtain two-dimensional (2D) texture images of snow microstructure [Brzoska, Lesaffre et al., 1999]. However, it is time-consuming and the voxel size is not small enough. Recently, X-ray microtomography has been utilized to obtain 3D high-resolution images of snow microstructure [Coleou, Lesaffre et al., 2001], which are more accurate than thin section images. Based on digitized 3D snow microstructures, micromechanics and heat transfer problems have been investigated by finite element analysis[Pieritz, Brzoska et al., 2004; Schneebeli, 2004; Kaempfer, Schneebeli et al., 2005].

The complexity and restrictions of these experimental methods provide the impetus to study an alternative one — stochastic reconstruction method. This method only needs a series of 2D surface images of snow microstructure. 3D microstructure can be

stochastically reconstructed from statistical characteristics extracted from 2D surface images.

The most widely studied stochastic reconstruction technique is the Gaussian filtering method [Joshi, 1974; Quiblier, 1984; Adler, Jacquin et al., 1990] which uses linear and nonlinear filters on the Gaussian random fields (GRFs) to match the correlation function in the reconstruction process. The method has been applied to porous or two-phase materials, such as sandstone, to calculate permeability.

Recently, the above stochastic reconstruction method has been extended and applied to the two-phase heterogeneous materials [Roberts and Teubner, 1995; Roberts, 1997; Roberts and Garboczi, 1999; Roberts and Garboczi, 2002]. The level-cut GRF model is used to model two-phase materials in this method. The relationship between the correlation function of GRFs and the two-point correlation function of the level-cut GRFs was derived [Teubner, 1991]. Using the 3D inverse fast Fourier transform, one can directly generate realizations of the target level-cut GRFs.

In the present study, we first obtained 3D high-resolution snow microstructure images by X-ray microtomography. Statistical characteristics were extracted from cross-sectional images. Then, the one-level-cut Gaussian random field model was used to statistically reconstruct snow microstructure from X-ray microtomography images. The comparison between the reconstructed microstructure and the real one showed a very good agreement.

Chapter 2 SNOW IMAGES BY X-RAY MICROTOMOGRAPHY

2.1 Introduction of computed tomography

The first clinical computed tomographic scanner was invented by engineer Godfrey Newbold Hounsfield in England at the Medical Systems Department of EMI Central Research Laboratories in the period of 1967 to 1972. His work was published in the British Journal of Radiology in 1973 [Hounsfield, 1973]. Independently, physicist Allan MacLeod Cormack laid the foundation for computed tomography (CT) when he worked on radiology and radiotherapy in Tufts University. Cormack published his theory in the Journal of Applied Physics in 1963 and 1964 [Cormack, 1963; Cormack, 1964]. There was little scientific or medical interest in his work until Hounsfield successfully obtained the first CT image in 1972. Hounsfield and Cormack were jointly awarded the Nobel Prize for Physiology or Medicine in 1979.

Computed tomography is a powerful nondestructive technique to obtain cross-sectional images of an object using special X-ray equipment. Characteristics of the internal structure of an object such as dimensions, shape, internal defects, and density are readily available from CT images. A schematic of a CT system is shown in Figure 1. The sample is placed on a rotary stage which is between an X-ray source and a camera. The rotary stage and the camera are controlled by a computer. The camera produces a two-dimensional projection image of the sample. Specialized computer software based on reconstruction algorithms makes it possible to produce cross-sectional images of the sample.

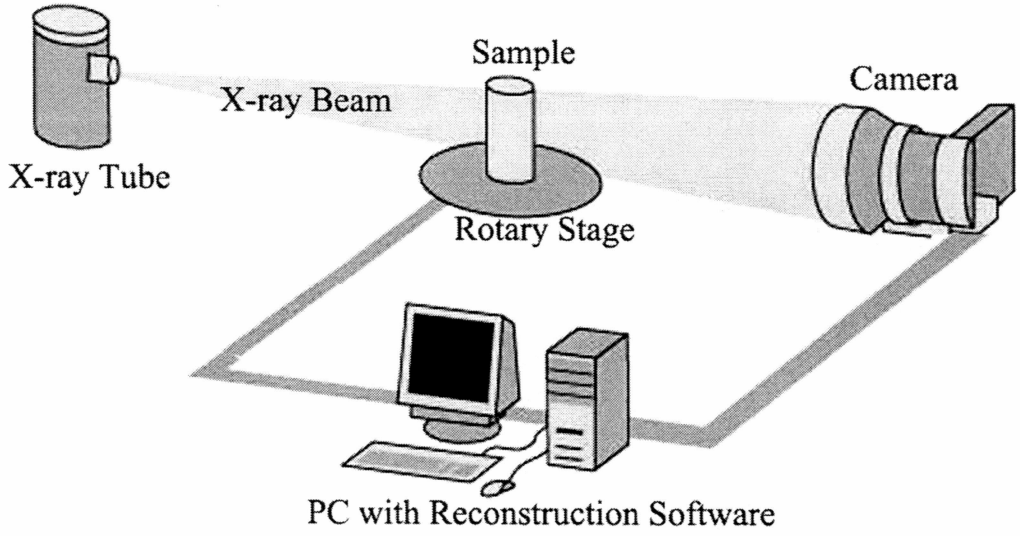

Figure 1 A schematic of a CT system

2.2 Acquisition and reconstruction

SkyScan 1172 Microtomograph shown in Figure 2 was used to acquire snow sample projection.

Figure 2 SkyScan 1172 Microtomograph

We collected natural snow from the Agricultural Field in Fairbanks, Alaska, in December 2006. The snow was sieved into different grain-sizes. The snow was allowed to sinter outside for a week and then stored in a freezer at a temperature of -30 Celsius. In May, 2007, the snow sample of the smallest grain-size (< 1 mm) was prepared from the sieved snow and put into a specially designed container which can be directly mounted onto the stage of the scanner. The snow sample along with a portable cooler was stored in the freezer for a week before it was scanned. We also made a bigger container for the larger grain-size snow sample. The small and big containers are shown in Figure 3 and Figure 4, respectively.

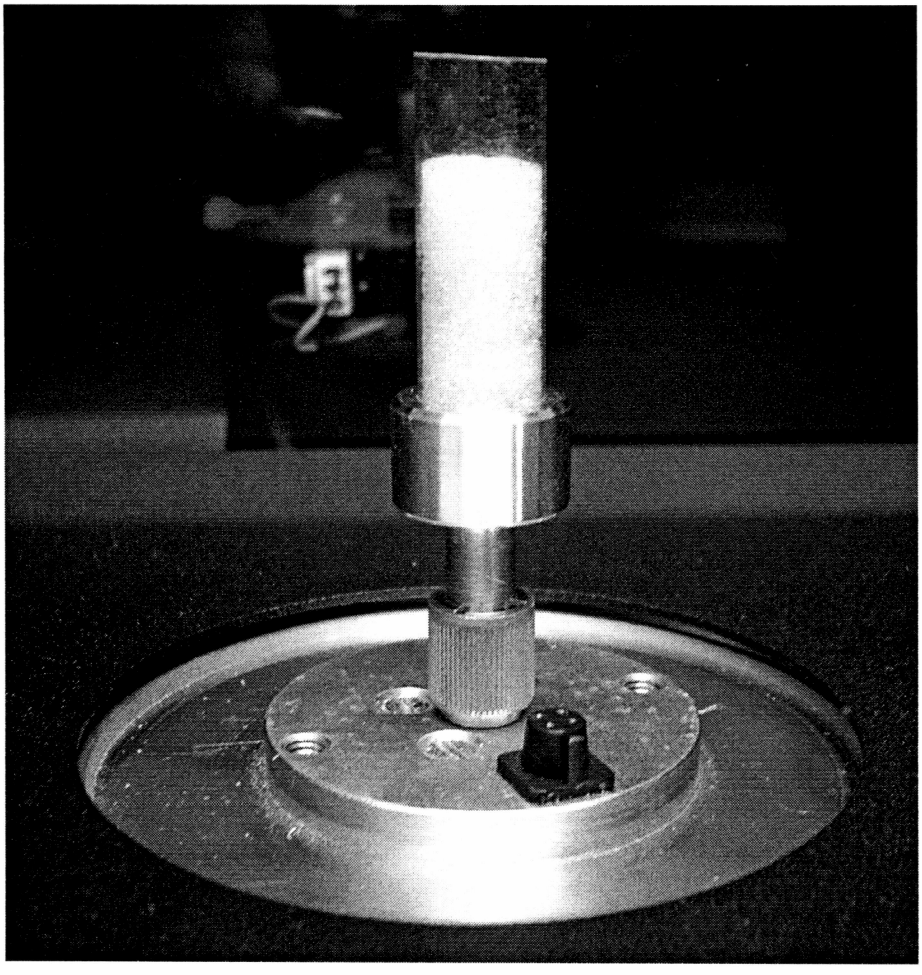

Figure 3 Small snow sample container, 10.5 mm in diameter

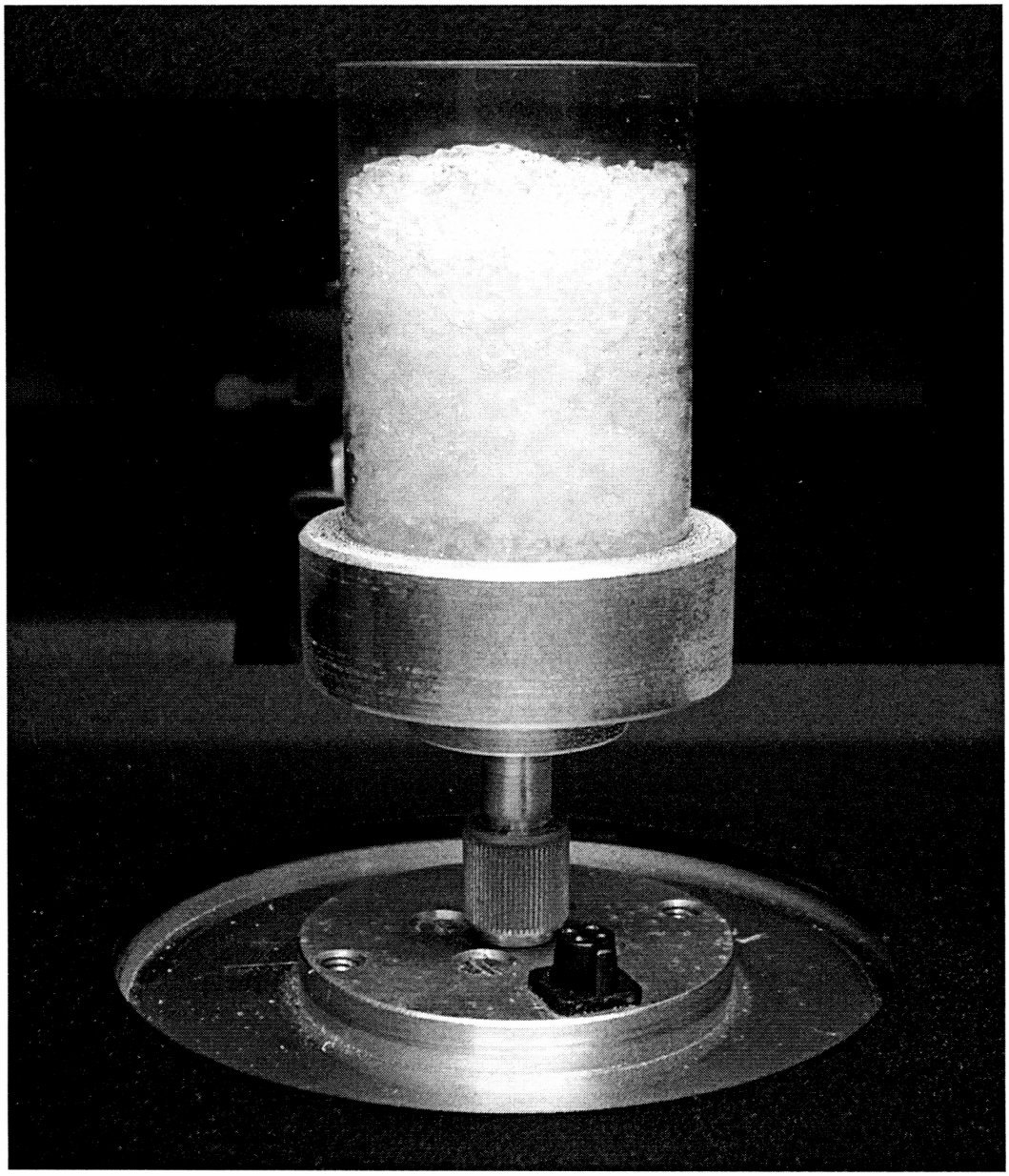

Figure 4 Big snow sample container, 30 mm in diameter

In order to obtain good results of snow microstructure, there are two major practical concerns in the scanning using our equipment and the cold-room facility. The first concern is to get a satisfactory profile line which is used to measure the brightness contrast of the projection image at the beginning of scanning; the voltage and exposure

time can be adjusted to achieve that. The second concern is to prevent snow from melting during the scanning process. The micro-CT scanner was taken into a cold room, where the temperature was kept below -10 Celsius, besides; the scan duration can not be too long due to X-ray energy accumulation. For the small grain-size snow sample which has small volume, decrease of the exposure time is very helpful to reduce the scan duration. However that does not apply to the bigger volume snow sample because long exposure time is needed to ensure good brightness contrast. In fact, it is not very easy to tell whether snow has melted or not with naked eyes if snow has just melted a little bit. One effective way to discern this is to check the results of reconstructed cross-sectional images. For instance, in Figure 5, one can see the "trace" which means the object has moved or deformed. Moreover, the ice grain boundary is blurred.

Figure 5 Snow melting while being scanned

For computed tomography, the nominal resolution is defined as the pixel size. The snow sample was scanned using two different nominal resolutions, named JOB1 and JOB2. One of the projection images of JOB1 is shown in Figure 6. Some of the acquisition settings are listed in Table 1.

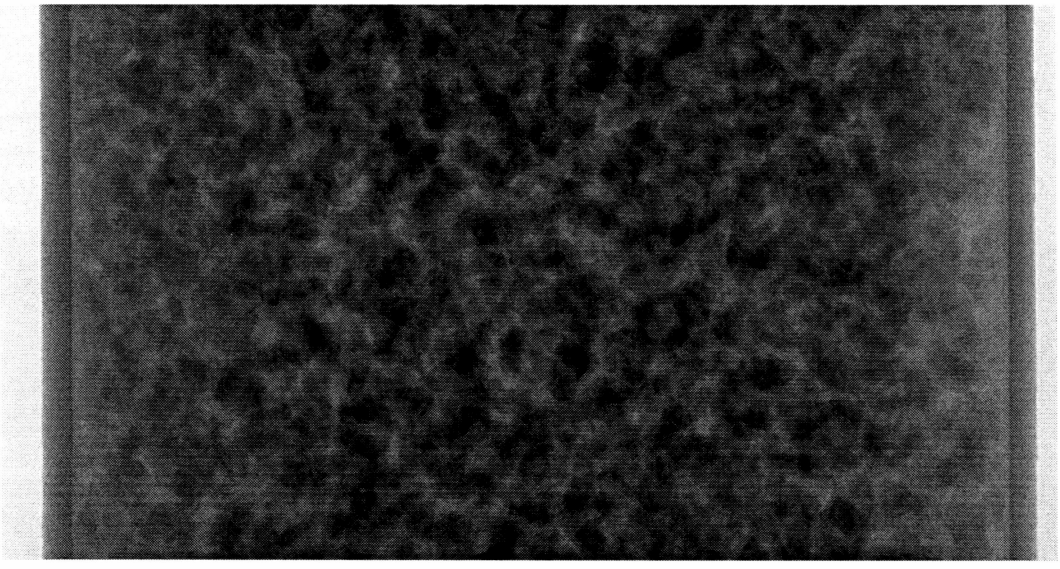

Figure 6 A projection image of JOB1, small grain-size snow sample

The reconstruction of snow microstructure from projection images was performed using the software NRecon (version 1.4.4). Some reconstruction settings are also listed in Table 1. Two reconstructed grey-level cross-sectional images of JOB1 and JOB2 are shown in Figure 7 and Figure 8 where the brighter pixels represent ice. We can see that there is a little more noise in Figure 7 than in Figure 8. The small nominal resolution and short exposure time are the reason for the low image quality.

Table 1 Acquisition and reconstruction settings

Scan Job Name	JOB1	JOB2
[Acquisition Settings]		
Source Voltage (kV)	80	80
Source Current (μA)	100	100
Camera Resolution	524*1000	1048*2000
Image Pixel Size (micron)	12	6
Filter	No filter	No filter
Exposure (ms)	79	147
Rotation Step (deg)	0.7	0.4
Frame Averaging	ON (5)	ON (5)
Random Movement	OFF (10)	OFF (10)
Use 360 Rotation	NO	NO
Scan duration	6 minutes	26 minutes
[Reconstruction Settings]		
Post alignment	-10.5	-21.5
Sections Count	468	935
Smoothing	1	1
Ring Artifact Correction	4	4
Object Bigger than Field of View	OFF	OFF

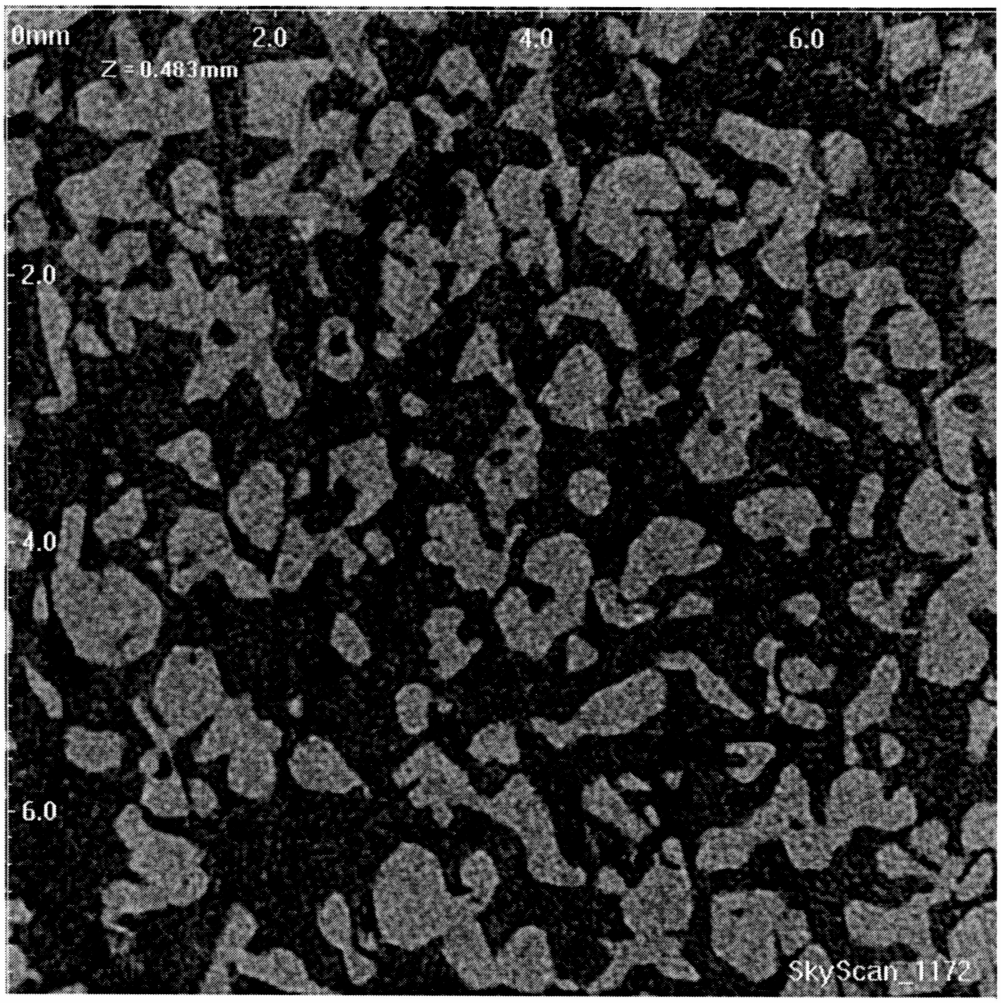

Figure 7 A reconstructed grey-level cross-sectional image of JOB1, image size 7.344 mm by 7.344 mm, image resolution 612 by 612

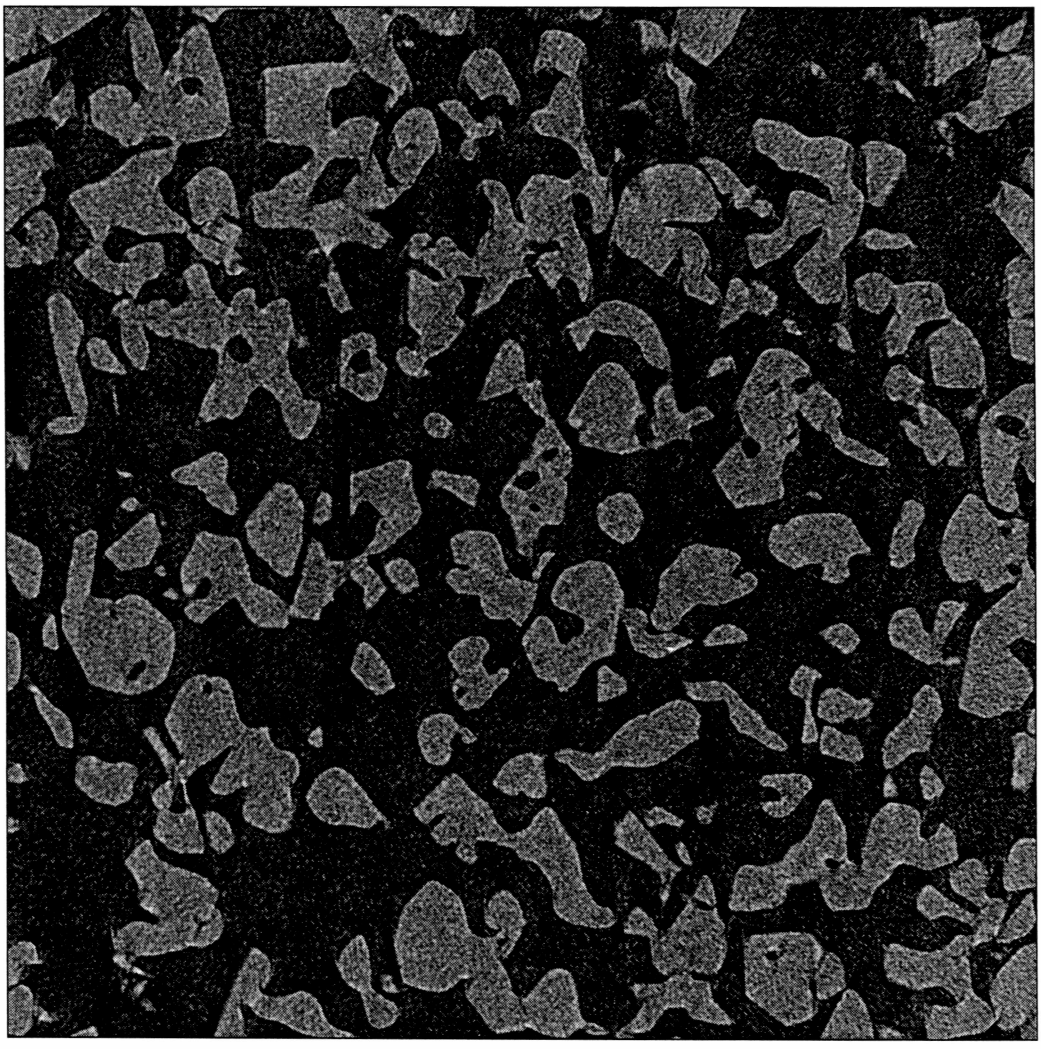

Figure 8 A reconstructed grey-level cross-sectional image of JOB2, image size 7.344 mm by 7.344 mm, image resolution 1225 by 1225

2.3 Binary images and 3D visualization

In order to obtain snow microstructure which is porous or two-phase, grey-level cross-sectional images must be converted into binary (black and white) ones. Since the grey-level images showed sufficient contrast, a threshold chosen from the histogram of the grey-level image was applied. For example, the grey-level pixel value histogram of JOB2

(calculated from all the cross-sectional images) is shown in Figure 9. The threshold is chosen to be the pixel value corresponding to the minimum value between two peaks, which is 56. With the chosen threshold, all cross-sectional images can be converted into binary images, of which black pixels represent ice and white pixels are pore. The binarized result of Figure 8 is shown in Figure 10.

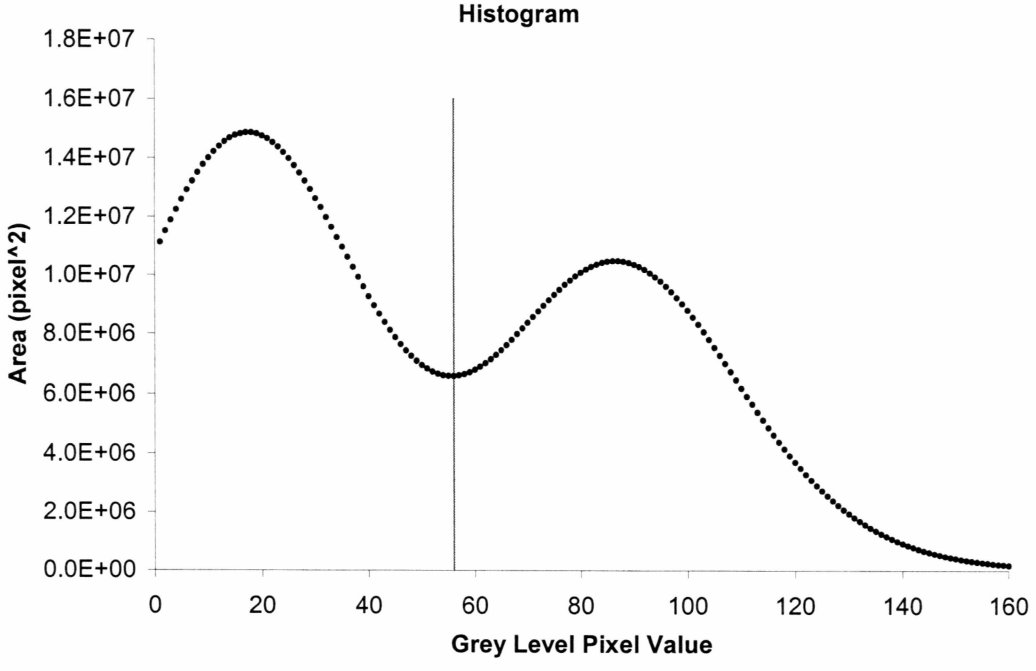

Figure 9 Histogram of JOB2

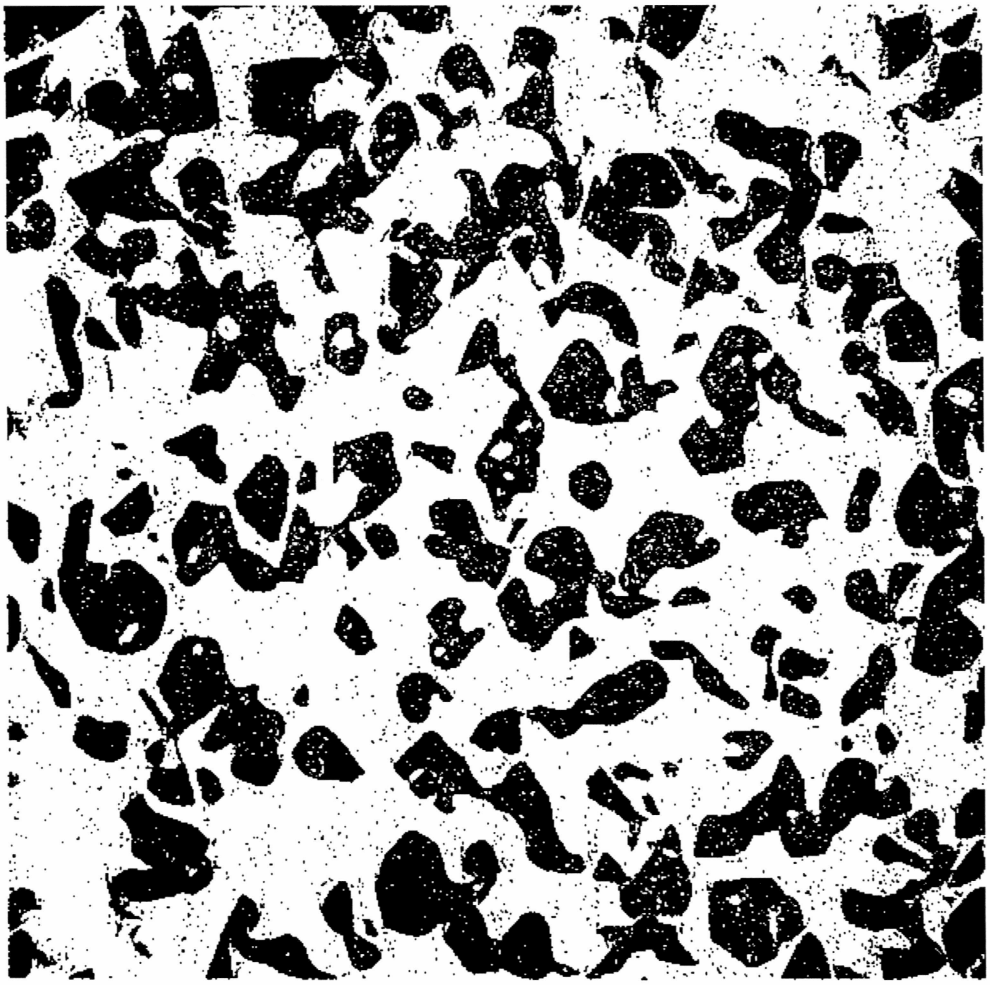

Figure 10 The binarized result of Figure 8

There are some unconnected ice particles in the binarized cross-sectional images due to both scanning noise and reconstruction procedure, which needed be removed. The software CTAn from the equipment manufacturer is used to fulfill this task. It searches unconnected ice parts in 3D space and then removes them, it also removes pores inside the ice particles. The result is shown in Figure 11.

Figure 11 Remove artifacts

With all cross-sectional images, the software CTAn was used to create a 3D visualization model with output as .p3g file. The software ANT was used to open .p3g file to view 3D microstructure of snow. Figure 12 shows a cube of the size $3.618 \times 3.618 \times 3.618$ mm^3. Figure 13 shows a cube of size $1.809 \times 1.809 \times 1.809$ mm^3.

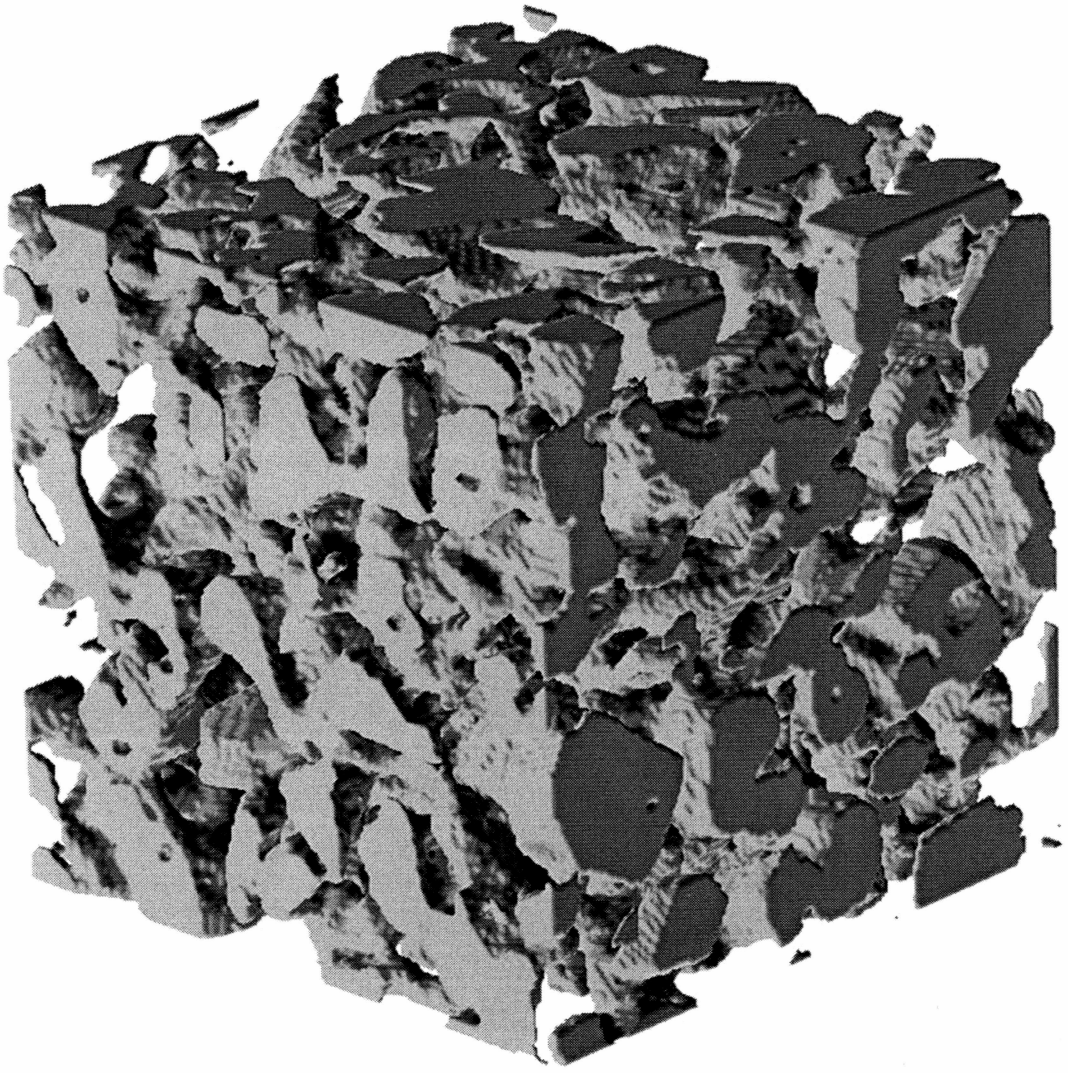

Figure 12 3D visualization of a cube of snow microstructure, side length 3.618 mm

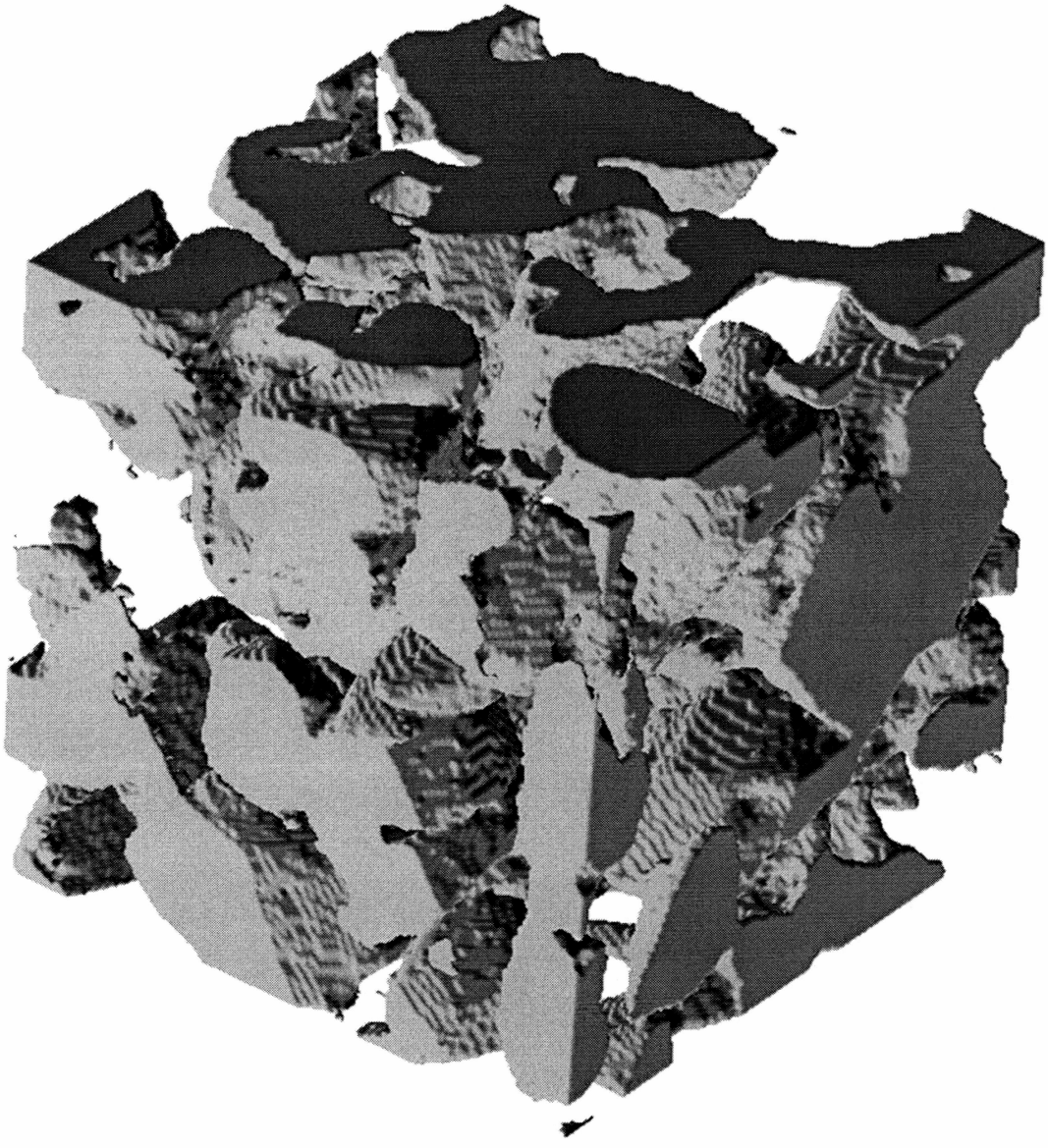

Figure 13 3D visualization of a cube of snow microstructure, side length 1.809 mm

2.4 Statistical characteristics of the snow sample

In this section, the porosity and two-point correlation function are extracted from micro-CT images of the small grain-size snow sample. These statistical characteristics are used as input to the statistical reconstruction procedure discussed in the next chapter.

Each binary cross-sectional image can be described as a two-dimensional matrix which takes the value 0 or 1 in the solid or pore phases, respectively. This can be represented by the indicator function:

$$Z(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \text{ belongs to the pore space} \\ 0 & \text{otherwise} \end{cases}. \tag{2-1}$$

The porosity p_{expt} and two-point correlation function $p_{\text{expt}}^{(2)}(\mathbf{r})$ are defined by the statistical average, which are denoted by brackets:

$$p_{\text{expt}} = \langle Z(\mathbf{x}) \rangle, \tag{2-2}$$

$$p_{\text{expt}}^{(2)}(\mathbf{r}) = \langle Z(\mathbf{x}) Z(\mathbf{r}+\mathbf{x}) \rangle, \tag{2-3}$$

where, $\mathbf{r}$ is the translation vector. For isotropic materials, $p_{\text{expt}}^{(2)}(\mathbf{r}) = p_{\text{expt}}^{(2)}(\|\mathbf{r}\|) = p_{\text{expt}}^{(2)}(r)$. Otherwise, the correlations for $\mathbf{r}$ parallel to the x-, y-, and z-axes are denoted $p_{\text{expt-x}}^{(2)}$, $p_{\text{expt-y}}^{(2)}$ and $p_{\text{expt-z}}^{(2)}$, respectively.

For the function $Z(\mathbf{x})$ defined in Equation (2-1), equation $Z^2(\mathbf{x}) = Z(\mathbf{x})$ holds. Then, we directly get the value of $p_{\text{expt}}^{(2)}(r)$ at two special points: $p_{\text{expt}}^{(2)}(0) = \langle Z^2(\mathbf{x}) \rangle = \langle Z(\mathbf{x}) \rangle = p_{\text{expt}}$, and $\lim_{r\to\infty} p_{\text{expt}}^{(2)}(r) \to \langle Z(\mathbf{x}) \rangle \langle Z(\mathbf{x}') \rangle = p_{\text{expt}}^2$. Due to these two values, we can see the physical meaning of $p_{\text{expt}}^{(2)}(r)$ that it represents the probability that two points a distance r apart will lie in the pore space.

On the other hand, the probability that two points a distance r apart will lie in the solid space is $p^{(2)}(r)_{solid} = \langle (1-Z(\mathbf{x}))(1-Z(\mathbf{x}+\mathbf{r})) \rangle = 1-2p_{\text{expt}} + p^{(2)}_{\text{expt}}(r)$.

Note that we only calculated the properties mentioned above for the small grain-size snow sample. The porosity p_{expt} of the small grain-size snow sample calculated by Equation (2-2) is 0.58502. The density of the small grain-size snow sample can be calculated from the porosity with the given ice density. The density is 0.917*(1-0.58502) g/cm^3 = 0.381 g/cm^3. Thus, the result of porosity can be verified by experimentally measuring the density of the snow sample. The measured density is 0.377 g/cm^3, which is very close to the calculated one.

The two-point correlation functions $p^{(2)}_{\text{expt-x}}$ and $p^{(2)}_{\text{expt-y}}$ measured from micro-CT cross-sectional images of JOB2 are shown in Figure 14, the two curves are almost the same (the maximum relative error is 1.6%), such that we can conclude that the snow sample is isotropic. Thus, radial function $p^{(2)}_{\text{expt}}(r_i)$ (r_i means discrete data points), instead of $p^{(2)}_{\text{expt-x}}$ and $p^{(2)}_{\text{expt-y}}$, is measured and used by the stochastic reconstruction hereafter.

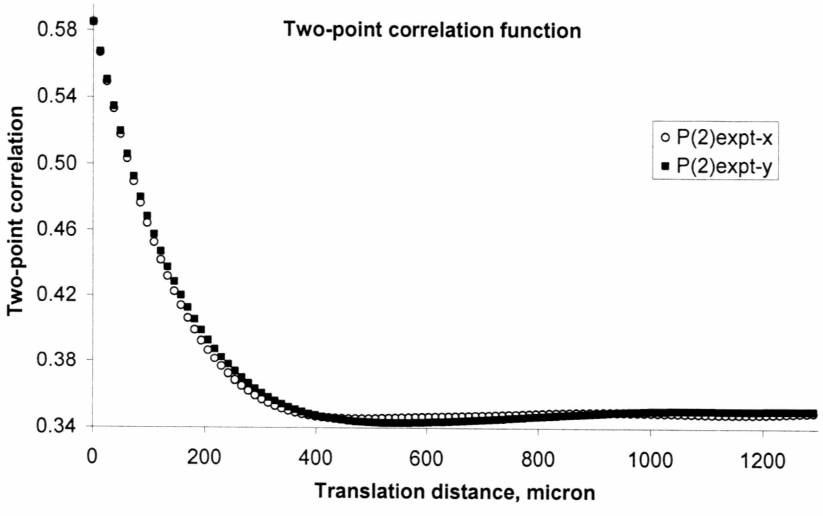

Figure 14 Two-point correlation functions $p^{(2)}_{\text{expt-x}}$ and $p^{(2)}_{\text{expt-y}}$ of JOB2

Theoretically speaking, the minimum value of the two-point correlation function $p_{\text{expt}}^{(2)}(r_i)$ is equal to $\left(p_{\text{expt}}\right)^2$ (see Section 3.1.2). For this reason, we plot the truncated $p_{\text{expt}}^{(2)}(r_i)$ in Figure 15.

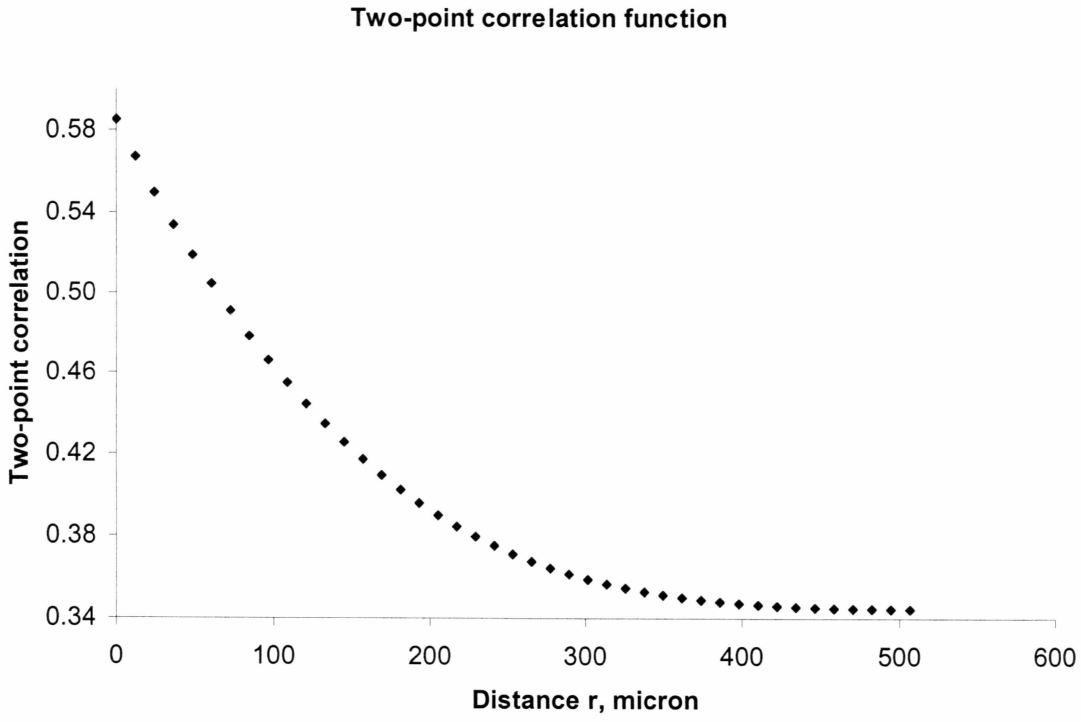

Figure 15 Two-point correlation function

Let $M(\mathbf{r}, L)$ denote a cubic measurement cell of side length L centered on the lattice vector $\mathbf{r}$. The local porosity in the measurement cell is defined as [Oren and Bakke, 2002]:

$$\psi(\mathbf{r}, L) = \frac{1}{\widehat{M}} \sum_{\mathbf{r}_i \in M(\mathbf{r}, L)} Z(\mathbf{r}_i) \tag{2-4}$$

where $\widehat{M}$ denotes the total number of voxels in the measurement cell. The local porosity for three different measurement cells is plotted in Figure 16. Local porosity is one statistical measure to estimate the size of statistically equivalent representative volume

elements. The change in local porosity for the measurement cell of side length 4.8 mm is less than 0.006, resulting in the change of 5.5 kg/m^3 for the snow density, which is a small portion compared with the snow density of 381 kg/m^3.

Note that the local porosity shown in Figure 16 is calculated from part of the snow sample, so the average of the local porosity is not the same as the one of the whole snow sample.

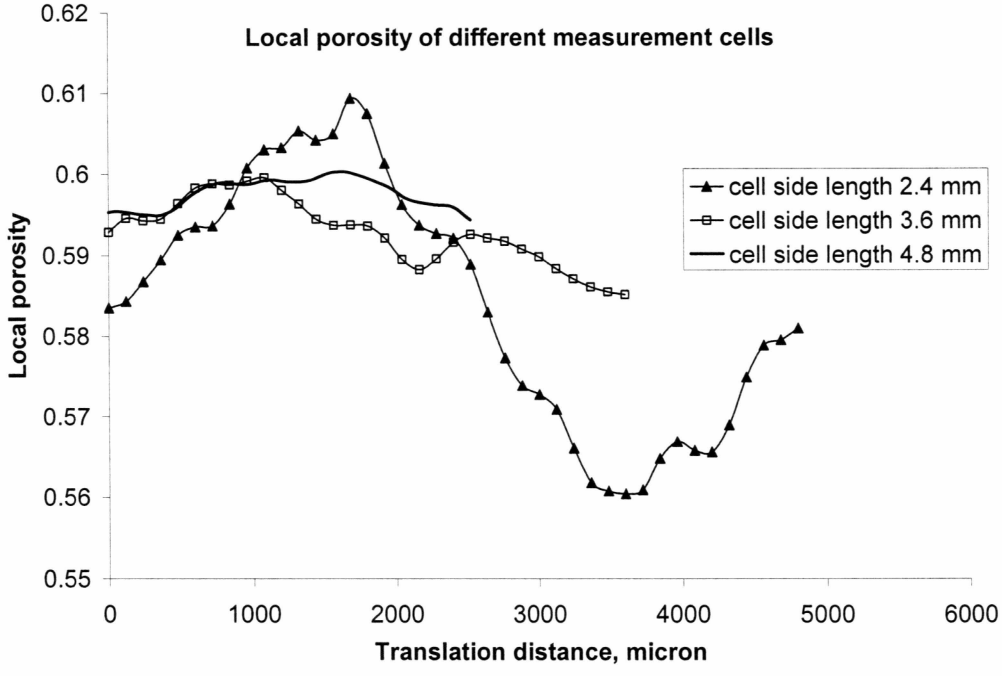

Figure 16 Local porosity of different measurement cells

Chapter 3 STOCHASTIC RECONSTRUCTION OF POROUS MATERIALS

3.1 Weak sense stationary Gaussian random fields

A probability space is specified by the three tuple (S,F,P), where $S(\omega)$ is called the sample space, F is Borel field specifying a set of events, and P is a probability measure allowing calculation of probabilities of all events.

A random variable $X(\omega)$ is defined as a real-valued function on S that satisfies the following: (a) $\{\omega : X(\omega) \le x\}$ is a member of F for all x, and (b) the probabilities of the events $\{\omega : X(\omega) = +\infty\}$ and $\{\omega : X(\omega) = -\infty\}$ are both zero.

A random process is a collection $\{X_t, t \in T\}$, where each X_t is a random variable, T is a time domain. For convenience, the random process $\{X_t, t \in T\}$ can be simply written as $X(t)$.

If, instead of considering a collection in the time domain, the collection of random variables $\{y_{\mathbf{r}}, \mathbf{r} \in V\}$ in the space domain is defined, then, we say a random field $y(\mathbf{r})$ is specified. The difference between a random process and a random field is the dimension of the parameter space; random processes refer to one dimensional parameter space, i.e., time domain, while random fields refer to two or three dimensional parameter space.

A random field $y(\mathbf{r})$ is a GRF if the random variables $y(\mathbf{r}_1), y(\mathbf{r}_2), \cdots, y(\mathbf{r}_n)$ are jointly Gaussian for all choices of $\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_n$ and for all n, which means that n-dimensional random variable $\mathbf{y} = [y(\mathbf{r}_1), y(\mathbf{r}_2), \cdots, y(\mathbf{r}_n)]^T$ has the joint probability density function:

$$f_n(y_1, y_2, \cdots, y_n) = ((2\pi)^n |\mathbf{G}|)^{-1/2} \exp(-\tfrac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^T \mathbf{G}^{-1}(\mathbf{y}-\boldsymbol{\mu})), \qquad (3\text{-}1)$$

where $(y_1, y_2, \cdots, y_n) = (y(\mathbf{r}_1), y(\mathbf{r}_2), \cdots, y(\mathbf{r}_n))$, $\boldsymbol{\mu} = (\mu_1, \mu_2, \cdots \mu_n)^T$ is the mean, and $\mathbf{G}$ is the covariance matrix.

A random field is weak sense stationary if:

(1) the expected value of the field $\langle y(\mathbf{r}) \rangle \equiv \mu$, which is independent of the space coordinate $\mathbf{r}$, and

(2) the correlation function $g(\mathbf{r}_1, \mathbf{r}_2) = \langle y(\mathbf{r}_1) y(\mathbf{r}_2) \rangle$ is a function of the distance $r = \|\mathbf{r}_1 - \mathbf{r}_2\|$ only, which means that $g(\mathbf{r}_1, \mathbf{r}_2)$ can be written as:

$$g(r) = \langle y(\mathbf{r}_1) y(\mathbf{r}_2) \rangle . \tag{3-2}$$

The snow microstructure to be reconstructed in this thesis is assumed to be statistically homogeneous and isotropic. Thus, for statistical reconstruction, we are dealing with an ergodic stationary GRF $y(\mathbf{r})$ only. In addition, $y(\mathbf{r})$ is a real-valued random field with $\mu = 0$ and $g(0) = 1$ throughout the thesis. In this case, Equation (3-1) can be written as:

$$f_n(y_1, y_2, \cdots, y_n) = ((2\pi)^n |\mathbf{G}|)^{-1/2} \exp(-\tfrac{1}{2} \mathbf{y}^T \mathbf{G}^{-1} \mathbf{y}) , \tag{3-3}$$

and $G_{ij} = g(\|\mathbf{r}_i - \mathbf{r}_j\|)$.

3.2 The Power spectral density function

In this thesis the 3D Fourier transform is used to generate a GRF $y(\mathbf{r})$ based on a given power spectral density function. As any random field is not square-integrable, we cannot apply the Fourier transform to $y(\mathbf{r})$ directly. For the purpose of defining the power spectral density function, a truncated function is defined as follows:

$$y_V(\mathbf{r}) = \begin{cases} y(\mathbf{r}) & \mathbf{r} \in V \\ 0 & \mathbf{r} \notin V \end{cases} , \tag{3-4}$$

where $V = (0, L) \times (0, L) \times (0, L)$ is the domain of interest. Now, the 3D Fourier transform can be applied for the truncated function $y_V(\mathbf{r})$:

$$Y_V(k_1,k_2,k_3)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} y_V(x_1,x_2,x_3)\exp\left(-2\pi i(k_1x_1+k_2x_2+k_3x_3)\right)dx_1dx_2dx_3\,, \quad (3\text{-}5)$$

On the other hand, we can write Y_V as:

$$Y_V(k_1,k_2,k_3)=a_V(k_1,k_2,k_3)+i\cdot b_V(k_1,k_2,k_3)\,. \quad (3\text{-}6)$$

The power spectral density function of random fields is defined as:

$$\rho(k_1,k_2,k_3)=\lim_{V\to\infty}\frac{\left\langle a_V^2(k_1,k_2,k_3)\right\rangle+\left\langle b_V^2(k_1,k_2,k_3)\right\rangle}{V}. \quad (3\text{-}7)$$

where, V also denotes the volume of the domain. Assuming $\left\langle a_V^2(k_1,k_2,k_3)\right\rangle=\left\langle b_V^2(k_1,k_2,k_3)\right\rangle$ [Roberts, 1997], substituting it into Equation (3-7) and rearranging gives:

$$\left\langle a_V^2(k_1,k_2,k_3)\right\rangle=\left\langle b_V^2(k_1,k_2,k_3)\right\rangle=\frac{1}{2}V\rho(k_1,k_2,k_3) \quad (3\text{-}8)$$

Here, the sign $\lim_{V\to\infty}$ is omitted.

The Wiener-Khinchin theorem [Torquato, 2001] states that the power spectral density of a weak sense stationary random process is the Fourier transform of the corresponding correlation function. For random fields in 3D space, the Wiener-Khinchin theorem has the following form:

$$\rho(k_1,k_2,k_3)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} g(\tau_1,\tau_2,\tau_3)\exp\left(-2\pi(k_1\tau_1+k_2\tau_2+k_3\tau_3)\right)d\tau_1d\tau_2d\tau_3\,, \quad (3\text{-}9)$$

where the correlation function $g(\tau_1,\tau_2,\tau_3)=\left\langle y(x_1,x_2,x_3)y(x_1+\tau_1,x_2+\tau_2,x_3+\tau_3)\right\rangle$.

As the snow microstructure in the present study is considered to be isotropic, it is assumed to have the same statistical characteristics in all directions. Thus, the correlation function $g(\tau_1,\tau_2,\tau_3)$ is a radial function which is a function of the distance from the origin only. The Fourier transform of a radial function is also radial; therefore, the power spectral density function $\rho(k_1,k_2,k_3)$ is a radial function. Based on these features, we write the correlation function and power spectral density function as

$g(\tau_1,\tau_2,\tau_3)=g(r)$ and $\rho(k_1,k_2,k_3)=\rho(k)$, where $r=\sqrt{(\tau_1^2+\tau_2^2+\tau_3^2)}$ and $k=\sqrt{(k_1^2+k_2^2+k_3^2)}$. For radial functions, the Fourier transform in spherical coordinates is only a one-dimensional integral transform [Torquato, 2001]:

$$\rho(k)=4\pi\int_0^\infty g(r)r^2\frac{\sin(2\pi kr)}{2\pi kr}dr. \tag{3-10}$$

The analytical correlation function used in this thesis is as follows [Roberts, 1997]:

$$g(r)=\frac{\exp(-r/\xi)-(r_c/\xi)\exp(-r/r_c)}{(1-r_c/\xi)}\left(\frac{\sin(2\pi r/d)}{2\pi r/d}\right), \tag{3-11}$$

where the three unknown parameters are correlation length ξ, cutoff scale r_c and domain scale d. Substituting Equation (3-11) into Equation (3-10), evaluating the definite integral yields:

$$\begin{aligned}\rho(k)=&\frac{d}{2\pi k}\frac{\xi^2}{(\xi-r_c)}\left(\frac{1}{1+4\pi^2\xi^2\left(d^{-1}-k\right)^2}-\frac{1}{1+4\pi^2\xi^2\left(d^{-1}+k\right)^2}\right)\\&+\frac{d}{2\pi k}\frac{r_c^2}{(r_c-\xi)}\left(\frac{1}{1+4\pi^2r_c^2\left(d^{-1}-k\right)^2}-\frac{1}{1+4\pi^2r_c^2\left(d^{-1}+k\right)^2}\right)\end{aligned}. \tag{3-12}$$

3.3 The one-level-cut Gaussian random field model

The one-level-cut GRF model is used to statistically reconstruct microstructure of porous materials in the present study.

Porous materials can be statistically reconstructed from a GRF $y(\mathbf{r})$, with $\mu=0$ and $g(0)=1$, by taking the region in space where $y(\mathbf{r})<\alpha$ as solid phase and the remaining region $y(\mathbf{r})\geq\alpha$ as pore. This is the so called the one-level cut GRF model. Here, α is the one-level-cut parameter. The phase function of this model, which is used to describe porous materials, is defined by Heaviside step function H as:

$$\phi(\mathbf{r}) = H\left(y(\mathbf{r}) - \alpha\right) = \begin{cases} 1, & y(\mathbf{r}) \geq \alpha, \text{ Pore} \\ 0, & y(\mathbf{r}) < \alpha, \text{ Solid} \end{cases}. \tag{3-13}$$

The n-point correlation function $p^{(n)}$ of this one-level-cut model is defined by the statistical average:

$$p^{(n)}(\mathbf{r}_1, \mathbf{r}_2, \cdots \mathbf{r}_n) = \left\langle \phi(\mathbf{r}_1)\phi(\mathbf{r}_2)\cdots\phi(\mathbf{r}_n) \right\rangle, \tag{3-14}$$

using Equation (3-1), the Equation above can be written as:

$$p^{(n)}(\mathbf{r}_1, \mathbf{r}_2, \cdots \mathbf{r}_n) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\cdots\int_{-\infty}^{+\infty} \phi(\mathbf{r}_1)\phi(\mathbf{r}_2)\cdots\phi(\mathbf{r}_n) f_n(y_1, y_2, \cdots, y_n) dy_1 dy_2 \cdots dy_n . \tag{3-15}$$

The statistically reconstructed microstructure in the thesis only shares $p^{(1)}$, which is the porosity p, and two-point correlation function $p^{(2)}$ with experimentally obtained one. For this reason, we write down their definitions separately:

$$p = \left\langle \phi(\mathbf{r}) \right\rangle, \tag{3-16}$$

$$p^{(2)}(r) = p^{(2)}(\mathbf{r}_1, \mathbf{r}_2) = \left\langle \phi(\mathbf{r}_1)\phi(\mathbf{r}_2) \right\rangle . \tag{3-17}$$

The porosity of this one-level-cut GRF model can be derived from Equations (3-15), (3-16) and (3-1):

$$p = (2\pi)^{-1/2} \int_{\alpha}^{\infty} e^{-t^2/2} dt . \tag{3-18}$$

The two-point correlation function $p^{(2)}(r)$ of this one-level-cut model is [Teubner, 1991]

$$p^{(2)}(r) = p - (2\pi)^{-1} \int_{g(r)}^{1} \frac{1}{\sqrt{1-t^2}} \exp\left(-\frac{\alpha^2}{1+t}\right) dt . \tag{3-19}$$

The above two equations, through which statistically reconstructed microstructures share the same statistical characteristics with the real one, are the essential equations used to fulfill statistical reconstruction.

In the process of verifying Equation (3-19), we came across an alternative derivation for the relationship between $p^{(2)}(r)$ and $g(r)$:

$$p^{(2)}(r)=\frac{\sqrt{1-g(r)^2}}{2\pi}\int_0^{\pi/2}\frac{\exp(w)\left(\exp(-u^2)-\sqrt{\pi}u\ \mathrm{erfc}(u)\right)}{1-g(r)\sin\hat{\theta}}d\hat{\theta}, \tag{3-20}$$

where, $w=\dfrac{\alpha^2\left(\sin\hat{\theta}-1\right)}{2\left(1-g(r)\sin\hat{\theta}\right)}$, $u=\alpha\sqrt{\dfrac{\left(1-g(r)\right)\left(1+\sin\hat{\theta}\right)}{2\left(1+g(r)\right)\left(1-g(r)\sin\hat{\theta}\right)}}$, the derivation is given in Appendix-1.

3.4 Generation of Gaussian random fields

There are several methods to generate stationary GRFs, such as:

(1) Fast Fourier transform (FFT) method.

(2) Summation of trigonometric series based on the spectral representation [Yamazaki and Shinozuka, 1988].

(3) Covariance matrix decomposition [Kozintsev, 1999].

The first method is used to generate GRFs in the present study due to efficiency. The FFT method is relatively easy to implement in MATLAB. In addition, by filtering out high frequencies, this method takes extremely short runtime even for large-scale 3D discrete GRFs.

The 3D inverse Fourier transform can be written as:

$$y_V(x_1,x_2,x_3)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}Y_V(k_1,k_2,k_3)\exp\left(2\pi i(k_1x_1+k_2x_2+k_3x_3)\right)dk_1dk_2dk_3\,. \tag{3-21}$$

The discrete Fourier transform (DFT) is applied to Equation (3-21) to compute $y_V(x_1,x_2,x_3)$ numerically. We take N^3 discrete values at uniformly-spaced points for the function $y_V(x_1,x_2,x_3)$ in the space domain V (which is a cube), with $h=L/N$ as the sample interval. We also take N^3 discrete values for $Y_V(k_1,k_2,k_3)$ within a cube in the frequency domain, with $\dfrac{1}{hN}$ as the sample interval and $\dfrac{1}{h}$ as the cube side length. It is

straightforward to choose $\left(\frac{-1}{2h},\frac{1}{2h}\right)\times\left(\frac{-1}{2h},\frac{1}{2h}\right)\times\left(\frac{-1}{2h},\frac{1}{2h}\right)$ as the DFT frequency domain and write the 3D inverse DFT based on Equation (3-21) as [Roberts, 1997]:

$$y_{lmn}=\frac{1}{N^3}\sum_{\gamma=-\widehat{N}}^{\widehat{N}}\sum_{\beta=-\widehat{N}}^{\widehat{N}}\sum_{\alpha=-\widehat{N}}^{\widehat{N}}\frac{Y_{\alpha\beta\gamma}}{h^3}\exp\left(\frac{2\pi i}{N}(\alpha l+\beta m+\gamma n)\right), \tag{3-22}$$

where $\widehat{N}=\frac{N-1}{2}$, y_{lmn} is the discretized form of y_V, $Y_{\alpha\beta\gamma}$ is the discretized form of Y_V. It is simple to write down the expression of $Y_{\alpha\beta\gamma}$ in Equation (3-22). With Equations (3-6) and (3-8), $Y_{\alpha\beta\gamma}$ has the expression as follows:

$$Y_{\alpha\beta\gamma}=(\text{randn}_1+i\cdot\text{randn}_2)\sqrt{\frac{V}{2}\rho\left(\frac{\sqrt{\alpha^2+\beta^2+\gamma^2}}{hN}\right)}, \tag{3-23}$$

where, MATLAB command "randn" means generating a random number with having mean zero and variance one, and randn_1 and randn_2 are independent random variables. In addition, $Y_{-\alpha,-\beta,-\gamma}=Y_{\alpha\beta\gamma}$ due to the condition that y_{lmn} is real.

There is no doubt that FFT algorithm should be used to numerically compute Equation (3-21). In the 3D case, the Cooley-Tukey FFT algorithm requires $3N^3\log_2 N$ operations to transform a 3D matrix of the size $N\times N\times N$ where N is a power of two. While the number of operations needed for direct matrix multiplication is N^6 which is $\frac{N^3}{3\log_2 N}$ times larger than the one for FFT. However, the indices in Equation (3-22) go from $-\widehat{N}$ to $\widehat{N}$, which is not the case for FFT in MATLAB. Though we can still apply FFT in MATLAB for Equation (3-22) using frequency shift technique, the computational time will increase greatly due to the additional frequency shift part. For this reason, we decide to choose $\left(0,\frac{1}{h}\right)\times\left(0,\frac{1}{h}\right)\times\left(0,\frac{1}{h}\right)$ as the DFT frequency domain which yields the expression of the 3D inverse FFT in MATLAB:

$$y_{lmn}=\frac{1}{N^3}\sum_{\gamma=0}^{N-1}\sum_{\beta=0}^{N-1}\sum_{\alpha=0}^{N-1}\frac{Y_{\alpha\beta\gamma}}{h^3}\exp\left(\frac{2\pi i}{N}(\alpha l+\beta m+\gamma n)\right). \tag{3-24}$$

Because the execution of FFT is fastest for those N values which are powers of two, we let N be even (best to be powers of two) hereafter, which make the execution of large-scale 3D FFT much faster. It should be noted that, for Equation (3-24), the expression of $Y_{\alpha\beta\gamma}$ is different from (3-23). The new expression is given below:

$$Y_{\alpha\beta\gamma}=\left(\text{randn}_1+i\cdot\text{randn}_2\right)\sqrt{\frac{V}{2}\rho\left(\frac{\sqrt{\left(\alpha-\widehat{H}\left(\alpha-\frac{N}{2}\right)N\right)^2+\left(\beta-\widehat{H}\left(\beta-\frac{N}{2}\right)N\right)^2+\left(\gamma-\widehat{H}\left(\gamma-\frac{N}{2}\right)N\right)^2}}{hN}\right)}, \tag{3-25}$$

where function $\widehat{H}$ is defined as follows:

$$\widehat{H}(x)=\begin{cases}1 & x>0\\ 0 & x\le 0\end{cases}. \tag{3-26}$$

$Y_{\alpha\beta\gamma}$ need to be subject to some restrictions such that y_{lmn} are real. We derive these conditions by looking into the 3D FFT expression:

$$\frac{Y_{\alpha\beta\gamma}}{h^3}=\sum_{n=0}^{N-1}\sum_{m=0}^{N-1}\sum_{l=0}^{N-1}y_{lmn}\exp\left(-\frac{2\pi i}{N}(\alpha l+\beta m+\gamma n)\right). \tag{3-27}$$

Noting that $\left(\exp(-2\pi i/N)\right)^0=\left(\exp(-2\pi i/N)\right)^N=1$, from Equation (3-27) we can get the following conditions:

$$\begin{cases}Y_{000}=\text{real}\\ Y_{\alpha\beta\gamma}=\overline{Y}_{(N-\alpha)(N-\beta)(N-\gamma)},\ (\alpha,\beta,\gamma=1,\cdots N-1)\\ Y_{0\beta\gamma}=\overline{Y}_{0(N-\beta)(N-\gamma)},\ (\beta,\gamma=1,\cdots N-1)\\ Y_{\alpha0\gamma}=\overline{Y}_{(N-\alpha)0(N-\gamma)},\ (\alpha,\gamma=1,\cdots N-1)\\ Y_{\alpha\beta0}=\overline{Y}_{(N-\alpha)(N-\beta)0},\ (\alpha,\beta=1,\cdots N-1)\\ Y_{\alpha00}=\overline{Y}_{(N-\alpha)00},\ (\alpha=1,\cdots N-1)\\ Y_{0\beta0}=\overline{Y}_{0(N-\beta)0},\ (\beta=1,\cdots N-1)\\ Y_{00\gamma}=\overline{Y}_{00(N-\gamma)},\ (\gamma=1,\cdots N-1)\end{cases}, \tag{3-28}$$

which are the necessary and sufficient conditions to ensure y_{lmn} to be real when using Equation (3-24) to compute y_{lmn}. In addition, to ensure that y_{lmn} is isotropic, we apply one more restriction to $Y_{\alpha\beta\gamma}$:

$$Y_{\alpha\beta\gamma}=0, \text{ if } \sqrt{\left(\alpha-\widehat{H}\left(\alpha-\frac{N}{2}\right)N\right)^2+\left(\beta-\widehat{H}\left(\beta-\frac{N}{2}\right)N\right)^2+\left(\gamma-\widehat{H}\left(\gamma-\frac{N}{2}\right)N\right)^2}>F_e. \quad (3\text{-}29)$$

where F_e is defined as the high-frequency threshold. To be isotropic, the maximum value of F_e is $N/2$. By changing F_e to a smaller value than $N/2$, higher frequency information can be filtered out, which can not only improve the smoothness of the grain boundaries of the reconstructed cross-sectional images, but also make the runtime of the stochastic reconstruction procedure extremely short.

3.5 Stochastic reconstruction procedure

In Equations (3-18) and (3-19), the porosity p and the two-point correlation function $p^{(2)}(r)$ can be measured from snow microstructure (see Section 2.4), the parameter α and correlation function $g(r)$ are sufficient to generate a realization of the one-level-cut GRF. Thus, Equations (3-18) and (3-19) are the "bridges" between real microstructures and statistically reconstructed ones. Therefore, the whole statistical reconstruction procedure is started from these two equations. Replacing p and $p^{(2)}(r)$ in these two equations with p_{expt} and $p^{(2)}_{\text{expt}}(r_i)$, which are the experimentally obtained porosity and two-point correlation function, yields:

$$p_{\text{expt}}=(2\pi)^{-1/2}\int_{\alpha}^{\infty}e^{-t^2/2}dt, \quad (3\text{-}30)$$

$$p^{(2)}_{\text{expt}}(r_i)=p_{\text{expt}}-(2\pi)^{-1}\int_{g(r_i)}^{1}\frac{1}{\sqrt{1-t^2}}\exp\left(-\frac{\alpha^2}{1+t}\right)dt. \quad (3\text{-}31)$$

By the numerical inversion of Equation (3-30) and (3-31), we can solve for level-cut parameter α and the discretized correlation function $g(r_i)$.

Using the error function $\mathrm{erf}(x) = 2\pi^{-1/2}\int_0^x \exp(-t^2)dt$, Equation (3-30) can be written as $p_{\mathrm{expt}} = (1/2)\left(1-\mathrm{erf}\left(\alpha/\sqrt{2}\right)\right)$, inverting it yields:

$$\alpha = \sqrt{2}\ \mathrm{erf}^{-1}\left(1-2p_{\mathrm{expt}}\right). \tag{3-32}$$

Here we discuss the detailed procedure to numerically invert Equation (3-31). First, we define a nonlinear function $I(g) = \int_g^1 \frac{1}{\sqrt{1-t^2}}\exp\left(-\frac{\alpha^2}{1+t}\right)dt$. Second, by substitution of integration variable: $t' = \sin^{-1} t$, $I(g)$ can be simplified as

$$I(g) = \int_{\sin^{-1} g}^{\pi/2} \exp\left(-\frac{\alpha^2}{1+\sin t'}\right)dt'. \tag{3-33}$$

Then, Equation (3-31) can then be written as:

$$I\left(g(r_i)\right) = 2\pi\left(p_{\mathrm{expt}} - p^{(2)}_{\mathrm{expt}}(r_i)\right), \tag{3-34}$$

which is a nonlinear equation that can be solved by numerical methods (the MATLAB program is given in Appendix-4).

With the correlation function $g(r_i)$, through a nonlinear curve fitting procedure, three unknown parameters ξ, r_c, d in Equation (3-11) can be determined, then, $\rho(k)$ is given by Equation (3-12). Finally, y_{lmn} can be generated using Equations (3-25) and (3-24). The statistical reconstruction procedure can be briefly illustrated as:

$$\boxed{p_{\mathrm{expt}}, p^{(2)}_{\mathrm{expt}}(r_i)} \rightarrow \boxed{\alpha, g(r_i)} \rightarrow \boxed{\xi, r_c, d\ \ (\text{i.e. } \rho(k))} \rightarrow \boxed{Y_{\alpha\beta\gamma}} \rightarrow \boxed{y_{lmn}} \rightarrow \boxed{\phi(\mathbf{r}_{lmn})},$$

where $\phi(\mathbf{r}_{lmn})$ denotes the value of $\phi(\mathbf{r})$ at discrete space points. More detailed procedure is given as follows:

Step 1: Compute p_{expt} and $p^{(2)}_{\text{expt}}(r_i)$ using Equations (2-2) and (2-3).

Step 2: Solve for α and $g(r_i)$ using Equations (3-32) and (3-34).

Step 3: Solve for ξ, r_c, d in Equation (3-11) by curve fitting.

Step 4: Numerically generate $Y_{\alpha\beta\gamma}$ using Equation (3-25).

Step 5: Compute y_{lmn} using 3D inverse FFT, i.e. Equation (3-24).

Step 6: Get $\phi(\mathbf{r}_{lmn})$ from y_{lmn} using Equation (3-13) (i.e. one-level-cut).

Chapter 4 RECONSTRUCTION RESULTS

In this chapter, the stochastic reconstruction procedure was performed to reconstruct snow microstructure from measured statistical information of the snow sample.

The one-level-cut parameter α computed using Equation (3-32) is:

$$\alpha = \sqrt{2}\ \mathrm{erf}^{-1}\left(1-2p_{\mathrm{expt}}\right) = \sqrt{2}\ \mathrm{erf}^{-1}\left(1-2\times 0.58502\right) = -0.21475. \tag{4-1}$$

The discrete correlation function $g(r_i)$ is obtained by numerically solving the nonlinear Equation (3-34). Parameters ξ, r_c and d in Equation (3-11) are determined by nonlinear least-square curve fitting:

$$\begin{cases} \xi = & 112.1 \\ r_c = & 95.6 \\ d = & 1000.8 \end{cases} \text{, unit: micron.} \tag{4-2}$$

The fitted analytical correlation function $g(r)$ expressed by Equation (3-11) and the discrete one $g(r_i)$ are plotted in Figure 17 for comparison.

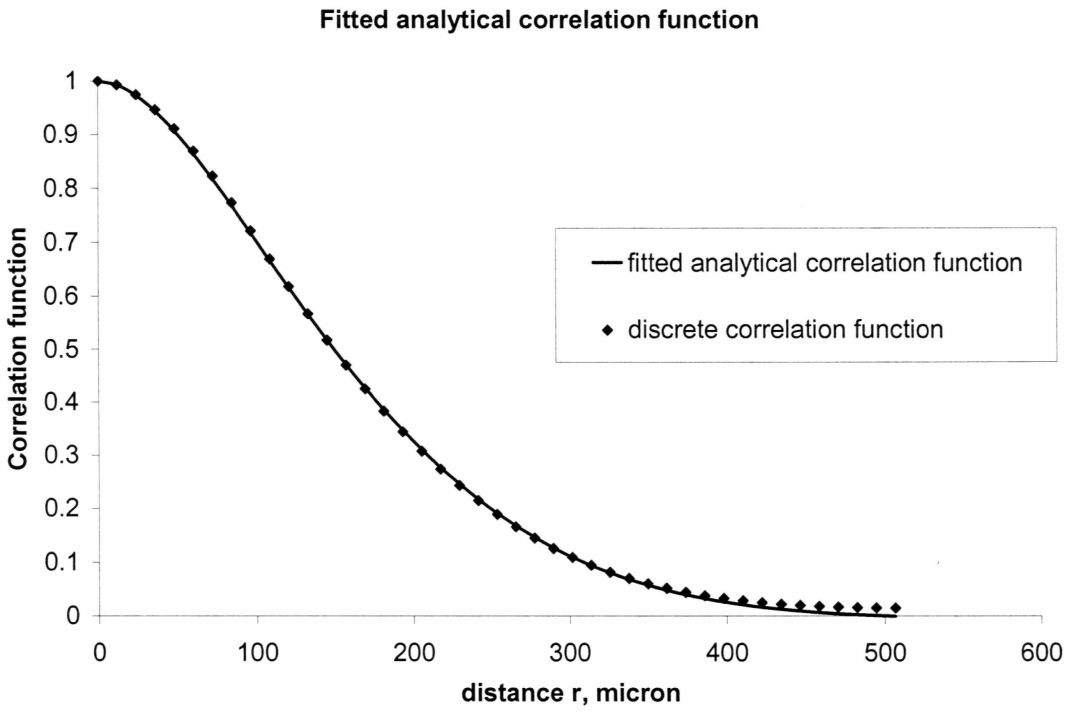

Figure 17 Analytical correlation function

A cube of simulated snow microstructure (side length 3.84 mm, 320^3 voxels) was stochastically reconstructed by the procedure discussed in Chapter 3. Nine sequential cross-sectional images are shown in Figure 18. For comparison, nine sequential cross-sectional images of the real snow microstructure (the same size as the fictitious one) are given in Figure 19. The grain-shape and grain-size in Figure 18 and Figure 19 are very similar. The two-point correlation functions of the fictitious microstructure and the real one are plotted in Figure 20. They are very close. 3D visualization of reconstructed snow microstructure is shown in Figure 21.

The runtime of reconstructing a cube of 320^3 voxels on Linux Workstations (Dual Core AMD Opteron(tm) Processor 280) is about 4 hours. However, changing the high-frequency threshold F_e from 160 to 20, the runtime becomes 2 minutes. This is because

most of the runtime is used to generate 3D matrix $Y_{\alpha\beta\gamma}$. The reconstructed cross-sectional image, for which high frequency information is filtered out, is shown in Figure 22.

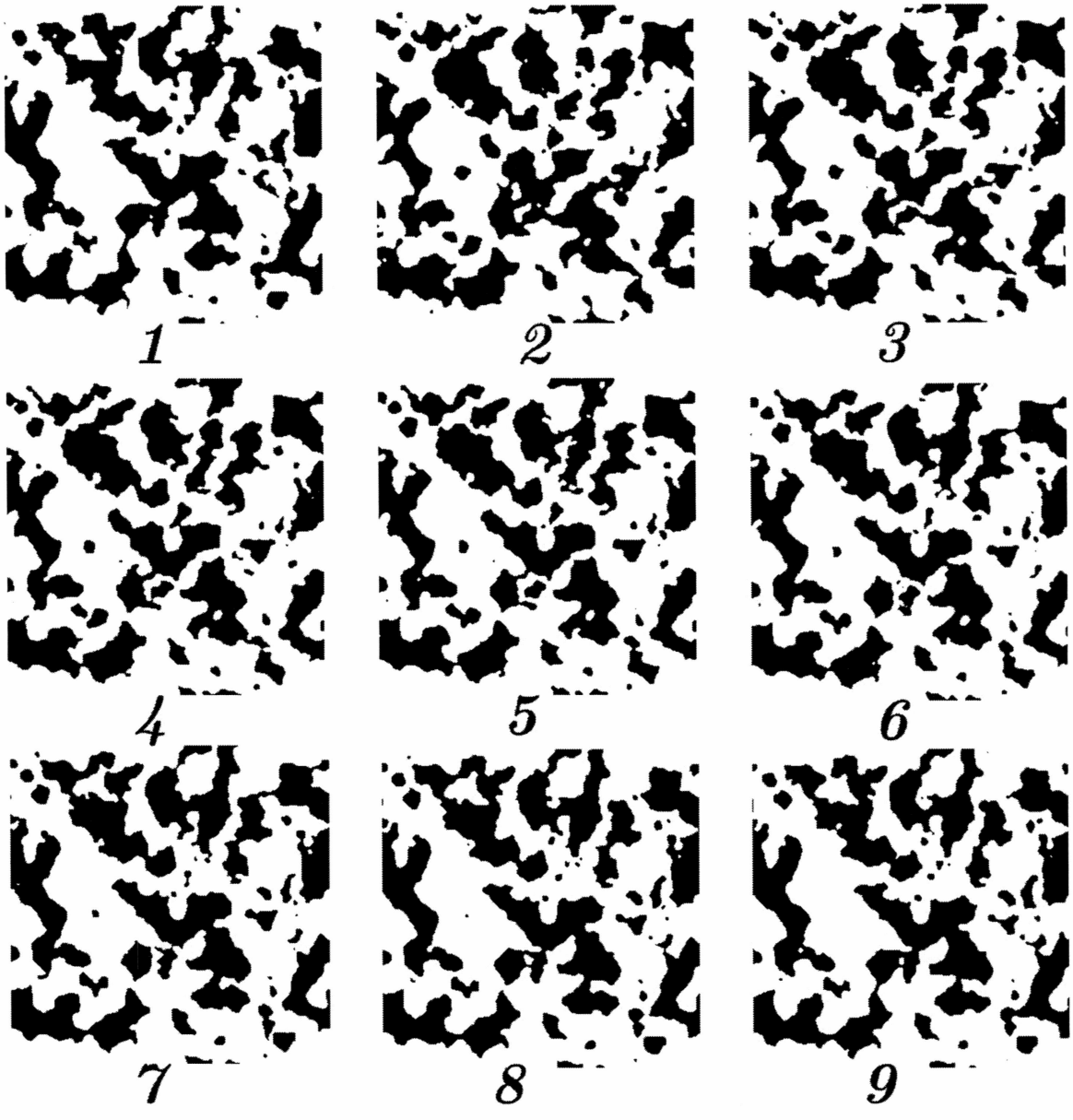

Figure 18 Cross-sectional images of stochastically reconstructed microstructure

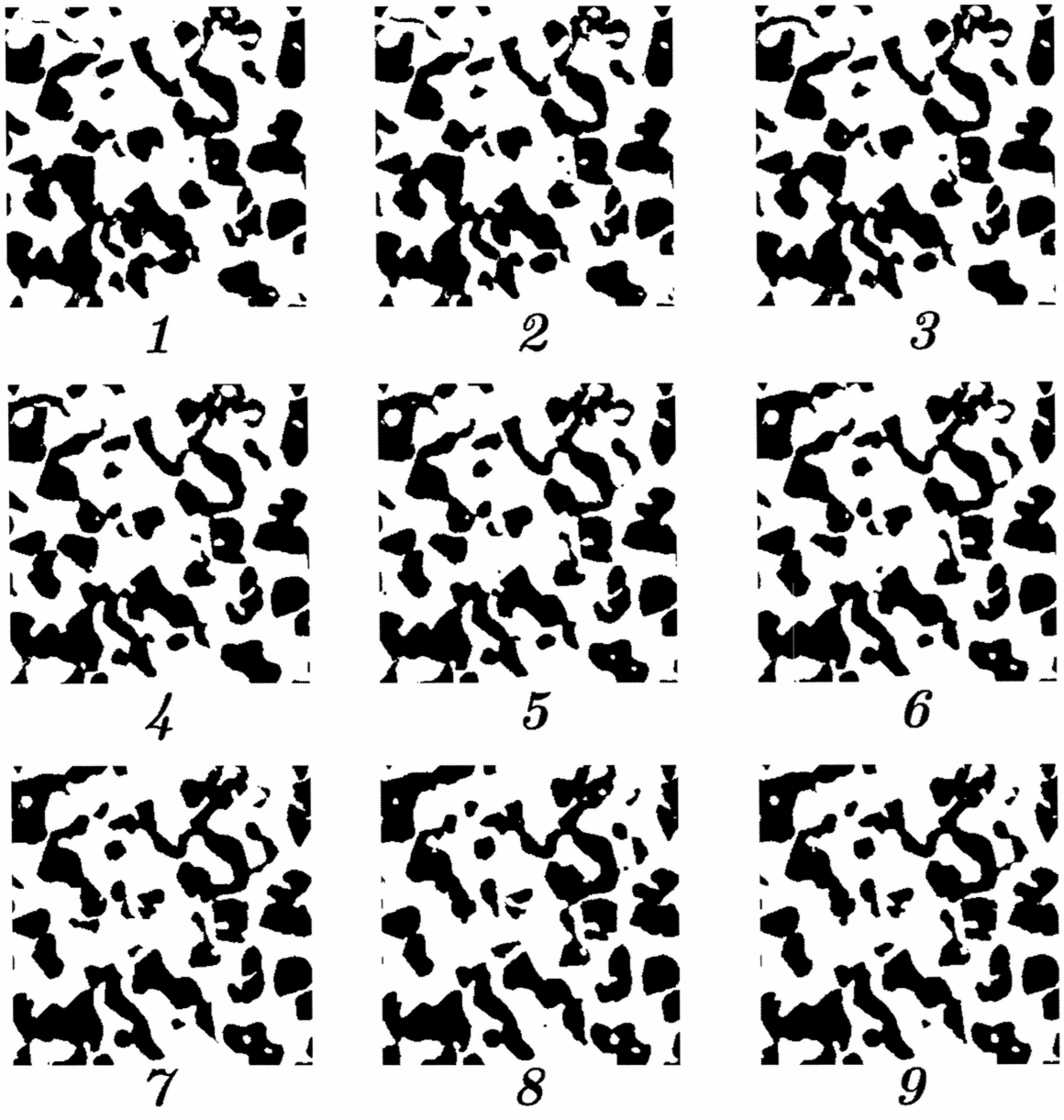

Figure 19 Cross-sectional images of the snow sample microstructure

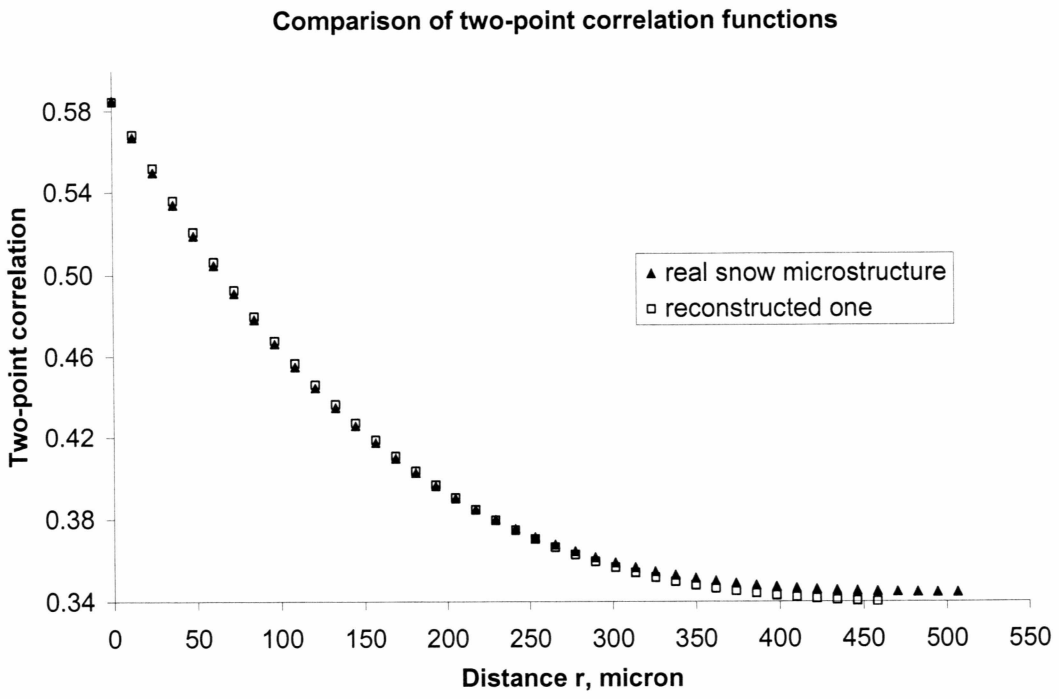

Figure 20 Comparison of two-point correlation function

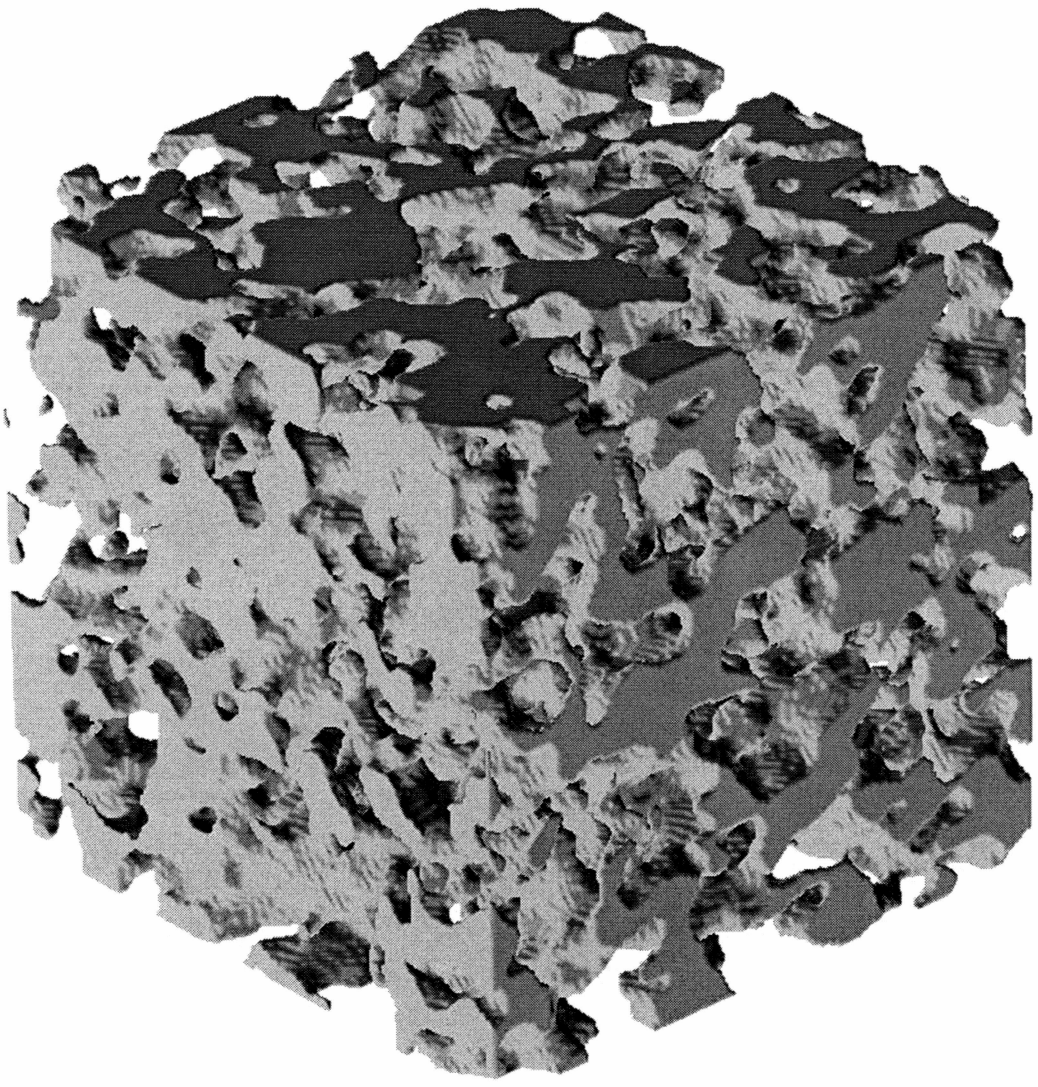

Figure 21 3D visualization of simulated snow microstructure, dimension: 3.840 mm by 3.840 mm by 3.840 mm

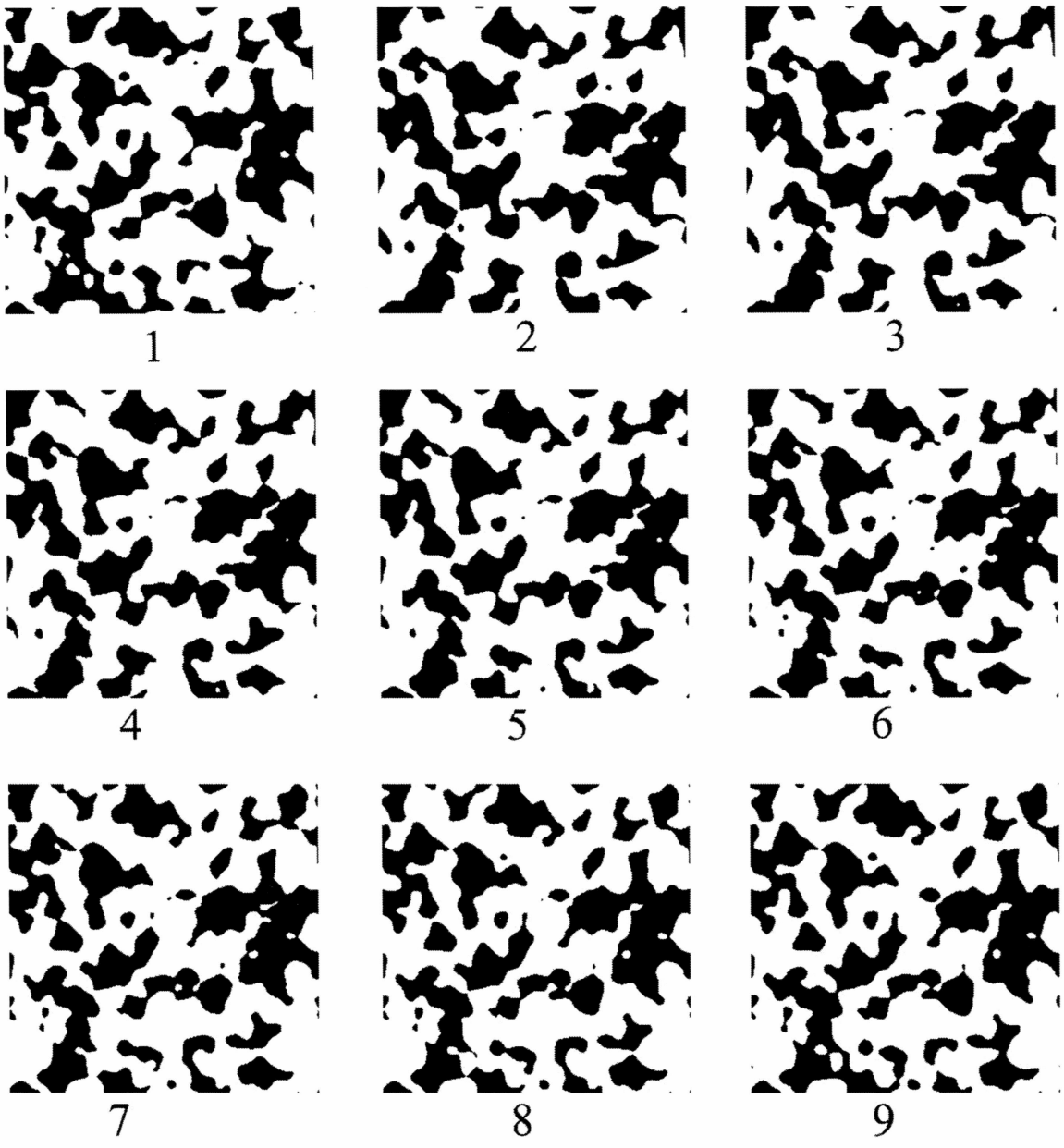

Figure 22 Results of high-frequency filtering

Chapter 5 CONCLUSIONS

The 3D high-resolution snow microstructure was obtained by X-ray microtomography. The experimental result was verified by measuring the density of the snow sample. Statistical characteristics were extracted from cross-sectional images. Among them, the porosity and two-point correlation function were used to stochastically reconstruct snow microstructure. Local porosity is very useful for estimating the size of statistically equivalent representative volume element.

Efficient computer programs have been developed in MATLAB for the whole stochastic reconstruction procedure, including the numerical inversion of the correlation function and the generation of 3D large-scale GRFs by 3D inverse fast Fourier transform. The procedure was applied to small grain-size (< 1 mm) snow sample. The quality of the reconstruction was assessed by comparing the two-point correlation function and cross-sectional images.

By changing the high-frequency threshold F_e to a value less than $N/2$, high frequency information can be filtered out. Not only can this high-frequency filter improve the smoothness of the grain boundary of the reconstructed microstructure, but also reduce the runtime of the stochastic reconstruction procedure by a large percentage. The effect of the high-frequency filter on the runtime is more significant, if the power spectral density function of the random microstructure is narrow.

Since the reconstruction method used in this thesis only shares first-order and second-order statistical information with the real microstructure, it can not ensure the simulated result is statistically equivalent to the real one because of the possibility of missing higher order information [Roberts, 1997]. The morphological quantities that depend on higher-order information are not taken into accounted by this method. However, from the point

of view of reconstruction, for certain kinds of microstructure, first- and second-order information are sufficient to reconstruct the fictitious microstructure having the same morphology and 3D macroscopic properties as the real one [Talukdar, Torsaeter et al., 2002]. From the reconstruction results in Chapter 4, we can conclude that this one-level-cut GRF model is suitable for the small grain-size snow sample.

REFERENCES

Adler, P. M., Jacquin, C. G. and Quiblier, J. A. (1990). "Flow in Simulated Porous-Media." International Journal of Multiphase Flow **16**(4): 691-712.

Brzoska, J. B., Lesaffre, B., Coleou, C., Xu, K. and Pieritz, R. A. (1999). "Computation of 3D curvatures on a wet snow sample." European Physical Journal-Applied Physics **7**(1): 45-57.

Coleou, C., Lesaffre, B., Brzoska, J. B., Ludwig, W. and Boller, E. (2001). Three-dimensional snow images by X-ray microtomography. Annals of Glaciology, Vol 32, 2001. Cambridge, Int Glaciological Soc. **32:** 75-81.

Cormack, A. M. (1963). "Representation of a Function by Its Line Integrals, with Some Radiological Application." Journal of applied physics **34**.

Cormack, A. M. (1964). "Representation of a Function by Its Line Integrals, with Some Radiological Application. II ." Journal of applied physics **35**.

Hounsfield, G. N. (1973). "Computerized transverse axial scanning (tomography). Part I: Description of system." British Journal of Radiology **46**: 1016-1022.

Johnson, J. B. and Hopkin, M. A. (2005). "Identifying microstructural deformation mechanisms in snow using discrete-element modeling." Journal of Glaciology **51**(174): 432-442.

Joshi, M. (1974). A class of stochastic models for porous materials, Ph.D. thesis, Univ. of Kansas.

Kaempfer, T. U., Schneebeli, M. and Sokratov, S. A. (2005). "A microstructural approach to model heat transfer in snow." Geophysical Research Letters **32**(21).

Klein-Paste, A., Sinha, N. K., Loset, S. and Norheim, A. (2007). "Microstructural analytical techniques for snow compacted by an aircraft tire." Tribology International **40**(2): 412-417.

Kozintsev, B. (1999). Computations with gaussian random fields, Ph.D. thesis. Department of Mathematics, University of Maryland at College Park.

Lee, J. H. and Liu, Q. (2005). "Upper Bound Indentation Theory for Snow-Depth Dependent Sinkage Calculation." Proceedings of the 15th International Conference of the ISTVS.

Lehning, M., Bartelt, P., Brown, B., Fierz, C. and Satyawali, P. (2002). "A physical SNOWTACK model for the Swiss avalanche warning Part II: Snow microstructure." Cold Regions Science and Technology **35**(3): 147-167.

Oren, P. E. and Bakke, S. (2002). "Process based reconstruction of sandstones and prediction of transport properties." Transport in Porous Media **46**(2-3): 311-343.

Pieritz, R. A., Brzoska, J. B., Flin, F. D., Lesaffre, B. and Coleou, U. (2004). From snow X-ray microtomograph raw volume data to micromechanics modeling: first results. Annals of Glaciology, Vol 38, 2004. Cambridge, Int Glaciological Soc. **38:** 52-58.

Quiblier, J. A. (1984). "A new three-dimensional modelling technique for studying porous media." Journal of Colloid and Interface Science **98**: 84-102.

Roberts, A. P. (1997). "Statistical reconstruction of three-dimensional porous media from two-dimensional images." Physical Review E **56**(3): 3203-3212.

Roberts, A. P. and Garboczi, E. J. (1999). "Elastic properties of a tungsten-silver composite by reconstruction and computation." Journal of the Mechanics and Physics of Solids **47**(10): 2029-2055.

Roberts, A. P. and Garboczi, E. J. (2002). "Elastic properties of model random three-dimensional open-cell solids." Journal of the Mechanics and Physics of Solids **50**(1): 33-55.

Roberts, A. P. and Teubner, M. (1995). "Transport-Properties of Heterogeneous Materials Derived from Gaussian Random-Fields - Bounds and Simulation." Physical Review E **51**(5): 4141-4154.

Schneebeli, M. (2004). Numerical simulation of elastic stress in the microstructure of snow. Annals of Glaciology, Vol 38, 2004. Cambridge, Int Glaciological Soc. **38:** 339-342.

Shapiro, L. H., Johnson, J. B., Sturm, M. and Blaisdell, G. L. (1997). Snow Mechanics Review of the State of Knowledge and Applications. CRREL Report.

Shoop, S. A. (2001). Finite Element Modeling of Tire-Terrain Interaction. CRREL Report.

Talukdar, M. S., Torsaeter, O., Ioannidis, M. A. and Howard, J. J. (2002). "Stochastic reconstruction of chalk from 2D images." Transport in Porous Media **48**(1): 101-123.

Teubner, M. (1991). "Level Surfaces of Gaussian Random-Fields and Microemulsions." Europhysics Letters **14**(5): 403-408.

Torquato, S. (2001). Random heterogeneous materials: microstructure and macroscopic properties, Springer.

Yamazaki, F. and Shinozuka, M. (1988). "Digital Generation of Non-Gaussian Stochastic Fields." Journal of Engineering Mechanics-Asce **114**(7): 1183-1197.

Zhang, T., Lee, J. H. and Liu, Q. (2005). "Finite Element Simulation of Tire-Snow Interaction under Combined Longitudinal and Lateral Slip Condition." Proceedings of the 15th International Conference of the ISTVS.

APPENDICES

Appendix-1 Derivation of Equation (3-20)

$$p^{(2)}(r) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \phi(\mathbf{r}_1)\phi(\mathbf{r}_2) f_2(y_1, y_2) dy_1 dy_2$$

$$= \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \phi(\mathbf{r}_1)\phi(\mathbf{r}_2)\left[\frac{1}{2\pi}(|\mathbf{G}|)^{-1/2}\exp(-\tfrac{1}{2}\mathbf{y}^T\mathbf{G}^{-1}\mathbf{y})\right] dy_1 dy_2$$

$$= \frac{1}{2\pi\sqrt{1-g^2}}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} H(y_1-\alpha)H(y_2-\alpha)\left(\exp\left(\frac{-1}{2(1-g^2)}\left(y_1^2+y_2^2-2y_1y_2g\right)\right)\right) dy_1 dy_2$$

$$= C_1\int_{\alpha}^{+\infty}\int_{\alpha}^{+\infty}\exp\left(-C_2\left(y_1^2+y_2^2-2y_1y_2g\right)\right) dy_1 dy_2$$

Here, $g = g(r)$, $C_1 = \dfrac{1}{2\pi\sqrt{1-g^2}}$, $C_2 = \dfrac{1}{2(1-g^2)}$.

Performing variable substitution: $\begin{cases} \hat{y}_1 = y_1 - \alpha \\ \hat{y}_2 = y_2 - \alpha \end{cases}$,

$$p^{(2)}(r) = C_1\int_{0}^{+\infty}\int_{0}^{+\infty}\exp\left(-C_2\left(\hat{y}_1^{\,2}+\hat{y}_2^{\,2}-2g\hat{y}_1\hat{y}_2+2\alpha(1-g)\left(\hat{y}_1+\hat{y}_2\right)+2\alpha^2(1-g)\right)\right) d\hat{y}_1 d\hat{y}_2 .$$

Performing variable substitution: $\begin{cases} \hat{y}_1 = \eta\cos\theta \\ \hat{y}_2 = \eta\sin\theta \end{cases}$,

$$p^{(2)}(r) = C_1 \int_0^{\frac{\pi}{2}} \int_0^{+\infty} \exp\left(-C_2\left(\eta^2 - g\eta^2 \sin 2\theta + 2\alpha\eta(1-g)(\cos\theta + \sin\theta) + 2\alpha^2(1-g)\right)\right)\eta d\eta d\theta$$

,

$$= \frac{\sqrt{1-g^2}}{2\pi} \int_0^{\frac{\pi}{2}} \frac{\exp(w)\left(\exp\left(-u^2\right) - \sqrt{\pi}\, u\, \mathrm{erfc}(u)\right)}{(1 - g\sin 2\theta)} d\theta$$

where, $u = \alpha\sqrt{\dfrac{(1-g)(1+\sin 2\theta)}{2(1+g)(1-g\sin 2\theta)}}$, $w = \dfrac{\alpha^2(\sin 2\theta - 1)}{2(1-g\sin 2\theta)}$.

Performing variable substitution: $\hat{\theta} = 2\theta$,

$$p^{(2)}(r) = \frac{\sqrt{1-g^2}}{2\pi} \int_0^{\frac{\pi}{2}} \frac{\exp(w)\left(\exp\left(-u^2\right) - \sqrt{\pi}\, u\, \mathrm{erfc}(u)\right)}{\left(1 - g\sin\hat{\theta}\right)} d\hat{\theta},$$

where, $u = \alpha\sqrt{\dfrac{(1-g)\left(1+\sin\hat{\theta}\right)}{2(1+g)\left(1-g\sin\hat{\theta}\right)}}$, $w = \dfrac{\alpha^2\left(\sin\hat{\theta} - 1\right)}{2\left(1-g\sin\hat{\theta}\right)}$.

Appendix-2 MATLAB program used for stochastic reconstruction

```
BeginTime = fix(clock);
SideLength = 320; % side length of reconstructed cube, Unit: scanned image pixel size
VN =320; % number of voxels along each cube side, mush be even, best be powers of 2
h = SideLength/VN; % sampling interval in space domain
V = SideLength^3; % volume of the cube
LevelCut = sqrt(2)*erfinv(1-2*0.58502*0.962);
ImageFolder = ['9SL' int2str(SideLength) 'VN' int2str(VN)];
mkdir(ImageFolder) % make a folder for images

% define inline power spectral density function
c = [9.3407    7.9670    83.4038]; % coefficients of Analytical CorrFunc
PSD          =          inline('8*pi/(c(1)-c(2))*(c(1)^4/(1+4*pi*pi*c(1)*c(1)*(1/c(3)-
fk)^2)/(1+4*pi*pi*c(1)*c(1)*(1/c(3)+fk)^2)-c(2)^4/(1+4*pi*pi*c(2)*c(2)*(1/c(3)-
fk)^2)/(1+4*pi*pi*c(2)*c(2)*(1/c(3)+fk)^2))','fk','c');

% define YIJK matrix
YIJK=zeros(VN,VN,VN);
EffectiveRadius = VN/2/SideLength;

% case1:  interior
for I=2:VN/2+1
  for J=2:VN/2+1
    for K=2:VN/2+1
      fk = sqrt((I-1)^2+(J-1)^2+(K-1)^2)/SideLength ;
      if fk <= EffectiveRadius
       YIJK(I,J,K)=(randn+i*randn)*sqrt(0.5*V*PSD(fk,c));
       YIJK(VN+2-I,VN+2-J,VN+2-K)=conj(YIJK(I,J,K));
```

```
    end
  end

  for K=VN/2+2:VN
    fk = sqrt((I-1)^2+(J-1)^2+(K-VN-1)^2)/SideLength ;
    if fk <= EffectiveRadius
     YIJK(I,J,K)=(randn+i*randn)*sqrt(0.5*V*PSD(fk,c));
     YIJK(VN+2-I,VN+2-J,VN+2-K)=conj(YIJK(I,J,K));
    end
  end
end

for J=VN/2+2:VN
  for K=2:VN/2+1
    fk = sqrt((I-1)^2+(J-VN-1)^2+(K-1)^2)/SideLength ;
    if fk <= EffectiveRadius
     YIJK(I,J,K)=(randn+i*randn)*sqrt(0.5*V*PSD(fk,c));
     YIJK(VN+2-I,VN+2-J,VN+2-K)=conj(YIJK(I,J,K));
    end
  end

  for K=VN/2+2:VN
    fk = sqrt((I-1)^2+(J-VN-1)^2+(K-VN-1)^2)/SideLength ;
    if fk <= EffectiveRadius
     YIJK(I,J,K)=(randn+i*randn)*sqrt(0.5*V*PSD(fk,c));
     YIJK(VN+2-I,VN+2-J,VN+2-K)=conj(YIJK(I,J,K));
    end
  end
end
```

```
end

% case2: I=1
for J=2:VN/2+1
  for K=2:VN/2+1
        fk = sqrt((J-1)^2+(K-1)^2)/SideLength ;
   if fk <= EffectiveRadius
    YIJK(1,J,K)=(randn+i*randn)*sqrt(0.5*V*PSD(fk,c));
    YIJK(1,VN+2-J,VN+2-K)=conj(YIJK(1,J,K));
   end
  end

  for K=VN/2+2:VN
        fk = sqrt((J-1)^2+(K-VN-1)^2)/SideLength ;
   if fk <= EffectiveRadius
    YIJK(1,J,K)=(randn+i*randn)*sqrt(0.5*V*PSD(fk,c));
    YIJK(1,VN+2-J,VN+2-K)=conj(YIJK(1,J,K));
   end
  end
end

% case3: J=1
for I=2:VN/2+1
  for K=2:VN/2+1
        fk = sqrt((I-1)^2+(K-1)^2)/SideLength ;
   if fk <= EffectiveRadius
    YIJK(I,1,K)=(randn+i*randn)*sqrt(0.5*V*PSD(fk,c));
    YIJK(VN+2-I,1,VN+2-K)=conj(YIJK(I,1,K));
   end
```

```
  end

  for K=VN/2+2:VN
        fk = sqrt((I-1)^2+(K-VN-1)^2)/SideLength ;
    if fk <= EffectiveRadius
     YIJK(I,1,K)=(randn+i*randn)*sqrt(0.5*V*PSD(fk,c));
     YIJK(VN+2-I,1,VN+2-K)=conj(YIJK(I,1,K));
    end
  end
end

% case4: K=1
for I=2:VN/2+1
  for J=2:VN/2+1
    fk = sqrt((I-1)^2+(J-1)^2)/SideLength ;
    if fk <= EffectiveRadius
     YIJK(I,J,1)=(randn+i*randn)*sqrt(0.5*V*PSD(fk,c));
     YIJK(VN+2-I,VN+2-J,1)=conj(YIJK(I,J,1));
    end
  end

  for J=VN/2+2:VN
    fk = sqrt((I-1)^2+(J-VN-1)^2)/SideLength ;
    if fk <= EffectiveRadius
     YIJK(I,J,1)=(randn+i*randn)*sqrt(0.5*V*PSD(fk,c));
     YIJK(VN+2-I,VN+2-J,1)=conj(YIJK(I,J,1));
    end
  end
end
```

```
% case5:I=J=1
  for K=2:VN/2
    fk = (K-1)/SideLength ;
    if fk <= EffectiveRadius
     YIJK(1,1,K)=(randn+i*randn)*sqrt(0.5*V*PSD(fk,c));
     YIJK(1,1,VN+2-K)=conj(YIJK(1,1,K));
    end
  end

% case6:I=K=1
  for J=2:VN/2
    fk = (J-1)/SideLength ;
    if fk <= EffectiveRadius
     YIJK(1,J,1)=(randn+i*randn)*sqrt(0.5*V*PSD(fk,c));
     YIJK(1,VN+2-J,1)=conj(YIJK(1,J,1));
    end
  end

% case7:K=J=1
  for I=2:VN/2
    fk = (I-1)/SideLength ;
    if fk <= EffectiveRadius
     YIJK(I,1,1)=(randn+i*randn)*sqrt(0.5*V*PSD(fk,c));
     YIJK(VN+2-I,1,1)=conj(YIJK(I,1,1));
    end
  end

% let 4 points value of YIJK be real number
```

```
% other 4 points are zero due to isotropy

fk = VN/2/SideLength ;
YIJK(1,1,VN/2+1)=randn*sqrt(V*PSD(fk,c));
YIJK(VN/2+1,1,1)=randn*sqrt(V*PSD(fk,c));
YIJK(1,VN/2+1,1)=randn*sqrt(V*PSD(fk,c));

YIJK(1,1,1)=randn*sqrt(V*PSD(0,c));

% solve for yLMN
yLMN = ifftn(YIJK); % 3D inverse FFT
clear YIJK;

% write one-level-cut GRF into cross-secional images
yLMN = sign(yLMN-LevelCut*h^3);

for I = 1:VN
  imwrite(yLMN(:,:,I),[ImageFolder '/' int2str(10000+I) '.pbm']);
end

% write the runtime to the file
EndTime = fix(clock);
fid = fopen([ImageFolder '/RunTime.txt'],'w+');
fprintf(fid,' %6i\n',EndTime-BeginTime);
fclose(fid);
```

Appendix-3 MATLAB program used to compute two-point correlation function

```
% compute porosity and discretized two-point correlation function of cross-sectional images

ImageBegin = 1;
ImageEnd = 320;

PreFix = '100';

VN =320; % image resolution

PixelSize = 6.03*2; % image pixel size, unit: micron

DistanceNum = round(VN/2); % selected distance to be calculated

TwoPointCorrX = zeros(1,DistanceNum+1);
TwoPointCorrY = zeros(1,DistanceNum+1);
Porosity = 0;

Counter = 0; % make sure all images are processed
for IN = ImageBegin : ImageEnd

   if IN <= 9
      ImageName = [PreFix '00' int2str(IN)];
   elseif IN <= 99
      ImageName = [PreFix '0' int2str(IN)];
   else
      ImageName = [PreFix int2str(IN)];
```

```
  end

  PixelValue = imread(ImageName, 'bmp');

  Porosity = Porosity + mean(mean(PixelValue)); % snow = 0, pore =1

% X direction two-point correlation function
  for Tao=0:DistanceNum
    temp1= PixelValue(:,1:VN-Tao).*PixelValue(:,Tao+1:VN);
    TwoPointCorrX(Tao+1) = TwoPointCorrX(Tao+1) + mean(mean(temp1));
  end

% Y direction two-point correlation function
  for Tao=0:DistanceNum
    temp1= PixelValue(1:VN-Tao,:).*PixelValue(Tao+1:VN,:);
    TwoPointCorrY(Tao+1) = TwoPointCorrY(Tao+1) + mean(mean(temp1));
  end

  Counter = Counter + 1;
end

% Average two point correlation function and porosity
TwoPointCorrX = TwoPointCorrX/Counter;
TwoPointCorrY = TwoPointCorrY/Counter;
Porosity = Porosity/Counter;

% Average X and Y direction
TwoPointCorr = (TwoPointCorrX + TwoPointCorrY)/2;
```

```
DistCoor = PixelSize*(0:1:DistanceNum);

% Output results
fidX = fopen('TwoPointCorrX.txt','w+');
fprintf(fidX,'%16.4e %16.4e\n',[DistCoor;TwoPointCorrX]);
fclose(fidX);

fidY = fopen('TwoPointCorrY.txt','w+');
fprintf(fidY,'%16.4e %16.4e\n',[DistCoor;TwoPointCorrY]);
fclose(fidY);

fidXY = fopen('TwoPointCorr.txt','w+');
fprintf(fidXY,'%16.4e %16.4e\n',[DistCoor;TwoPointCorr]);
fclose(fidXY);

fidPorosity = fopen('Porosity.txt','w+');
fprintf(fidPorosity,'%16.6f %16i',Porosity, Counter);
fclose(fidPorosity);
```

Appendix-4 MATLAB program used to compute correlation function

```
% load experimental 2-point correlation data
load Pexpt.txt
load P2expt.txt

DataSize = size(P2expt,1);

% get levelcut parameter
LevelCut = sqrt(2)*erfinv(1-2*Pexpt);

Pbar = 2*pi*(Pexpt-P2expt);

fun1 = inline('exp(-0.04611879506./(1+sin(x)))');

warning off MATLAB:quad:MinStepSize

MaxIteration = 10000;

for I = 1:DataSize
        gSmall = 0.0;
        gLarge = 1.0;
        g = (gSmall+gLarge)/2;

        for J = 1: MaxIteration
                temp = g;
                Intg = quad(fun1,asin(g),pi/2);
                if Intg < Pbar(I)
                        gLarge = g;
```

```
                g = (g+gSmall)/2;
            elseif Intg > Pbar(I)
                gSmall = g;
                g = (g+gLarge)/2;
            else
                CorrFunc(I) = g;
                break
            end

            if abs(temp-g)< 1.0e-6
                CorrFunc(I) = g;
                break
            end

 end
end

fid = fopen('CorrFunc.txt','w+');
fprintf(fid,' %16.7e\n',CorrFunc);
fclose(fid);

fid = fopen('LevelCut.txt','w+');
fprintf(fid,' %16.7f\n',LevelCut);
fclose(fid);
```

Appendix-5 MATLAB program used to compute local porosity

```
ImageBegin = 100;
LL = 100; % side length of measurement cell
ImageEnd = ImageBegin+LL-1;
PreFix = 'fine2_bin0';
VN =612; % image resolution
TSL = 3; % translating step length
LocalPor = zeros(fix((VN-LL+1)/TSL)+1);
Porosity = 0;
Counter = 0;

for IN = ImageBegin : ImageEnd

   if IN <= 9
      ImageName = [PreFix '00' int2str(IN)];
   elseif IN <= 99
      ImageName = [PreFix '0' int2str(IN)];
   else
      ImageName = [PreFix int2str(IN)];
   end

   PixelValue = imread(ImageName, 'bmp');

   Porosity = Porosity + mean(mean(PixelValue)); % snow = 0, pore =1

% compute local porosity

         for I =1: TSL: VN-LL+1
```

```
            for J=1: TSL: VN-LL+1
                I2 = fix(I/TSL)+1;
                J2 = fix(J/TSL)+1;
                LocalPor(I2,J2)=LocalPor(I2,J2)+ mean(mean(PixelValue(I:I+LL-
1,J:J+LL-1)))  ;
            end
        end

 Counter = Counter + 1;
end

% average local porosity
LocalPor = LocalPor/Counter;
Porosity = Porosity/Counter;
```

Appendix-6 MATLAB program used for nonlinear fitting

```
DataSizeP = size(CorrFuncP,1);
rP = (1:DataSizeP)';
c0 = [9.3407    7.9670    83.4038]; % Starting guess coefficients: xi,rc,d

AnaCorrFunc                =                inline('(exp(-r/c(1))-c(2)/c(1)*exp(-r/c(2)))/(1-
c(2)/c(1)).*sin(2*pi*r/c(3))./r/2/pi*c(3)','c','r');

[c,resnorm] = lsqcurvefit(AnaCorrFunc,c0,rP,CorrFuncP)
```